UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE MATEMÁTICA
Mestrado Profissional em Matemática em Rede Nacional

Ilso Francisco dos Santos

Algoritmos de Ordenação: uma abordagem didática para o Ensino Médio

RECIFE
2020

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE MATEMÁTICA
Mestrado Profissional em Matemática em Rede Nacional

Ilso Francisco dos Santos

Algoritmos de Ordenação: uma abordagem didática para o Ensino Médio

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Federal Rural de Pernambuco como requisito parcial para obtenção do título de Mestre em Matemática.

Orientador: Prof. Dr. Marcelo Pedro dos Santos

RECIFE
2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

S237a SANTOS, ILSO FRANCISCO DOS

Algoritmos de ordenação: uma abordagem didática para o ensino médio / ILSO FRANCISCO DOS SANTOS. - 2020.

78 f. : il.

Orientador: MARCELO PEDRO DOS SANTOS.

Inclui referências.

Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Mestrado Profissional em Matemática (PROFMAT), Recife, 2020.

1. Algoritmos de ordenação. 2. Parâmetros Curriculares. 3. Competências. 4. Habilidades. 5. Ensino Médio. I. SANTOS, MARCELO PEDRO DOS, orient. II. Título

CDD 510

ILSO FRANCISCO DOS SANTOS

Algoritmos de Ordenação: Uma Abordagem Didática para o Ensino Médio

Trabalho apresentado ao Programa de Mestrado Profissional em Matemática – PROFMAT do Departamento de Matemática da UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO, como requisito parcial para obtenção do grau de Mestre em Matemática.

Aprovado em 03 /07/2020

BANCA EXAMINADORA

Prof. Dr. Marcelo Pedro dos Santos (Orientador)– UFRPE

Prof. Dr. Eudes Naziazeno Galvão – DMat/UFPE

Profª. Drª. Tarciana Maria Santos da Silva – PROFMAT/UFRPE

À minha esposa e filhos
Aos meus pais
Aos meus colegas de turma
Aos meus colegas de trabalho
Aos meus professores
Ao meu orientador

Agradecimentos

Primeiramente a Deus por ter me proporcionado a oportunidade aos estudos.
À minha esposa Valéria e filhos: Iago e Aimê pela ajuda e companheirismo nos momentos de estudo e na ausência durante o curso.
Aos diretores das escolas que trabalho, pois sempre que possível se colocavam à disposição para ajudar administrativamente.
Aos meus professores do PROFMAT pela paciência e disponibilidade para ajudar nos momentos de estudo.
Ao meu orientador Marcelo Pedro que guiou com muita sabedoria o percurso da pesquisa.

"A verdadeira motivação vem da realização,
desenvolvimento pessoal, satisfação no trabalho e reconhecimento."
(Frederick Herzberg)

DECLARAÇÃO

Eu, **ILSO FRANCISCO DOS SANTOS** declaro, para devidos fins e efeitos, que a dissertação sob título **Algoritmos de Ordenação: uma abordagem didática para o Ensino Médio**, entregue como Trabalho de Conclusão de curso para obtenção do título de mestre, com exceção das citações diretas e indiretas claramente indicadas e referenciadas, é um trabalho original. Eu estou consciente que a utilização de material de terceiros incluindo uso de paráfrase sem a devida indicação das fontes será considerado plágio, e estará sujeito à processo administrativos da Universidade Federal Rural de Pernambuco e sanções legais. Declaro ainda que respeitei todos os requisitos dos direitos de autor e isento a Pós-graduação PROFMAT/UFRPE, bem como o professor orientador Marcelo Pedro dos Santos, de qualquer ônus ou responsabilidade sobre a sua autoria.

Recife, 20 de agosto de 2020.

____

ILSO FRANCISCO DOS SANTOS

Resumo

Baseados na nova Base Nacional Curricular Comum (BNCC), em especial, nas competências habilidades correlacionadas com algoritmos, faremos uma abordagem didática propondo atividades que possam ser trabalhadas no processo de ensino-aprendizagem. Apesar das competências e habilidades fazerem referências a algoritmos, iremos enfocar, especificamente, Algoritmos de Ordenação sem, no entanto, observar o modelo computacional dos mesmos. Analisaremos inicialmente os Parâmetros Curriculares Nacionais 5ª a 8ª séries(PCN-5ª a 8ª), Parâmetros Curriculares Nacionais do Ensino Médio(PCNEM) e seu complemento PCNEM_{+}, BNCC e também Parâmetros Curriculares da Educação Básica do Estado de Pernambuco, enfatizando que o trabalho com Algoritmos de Ordenação favorece não só a resolução de problemas, como o desenvolvimento de habilidades e competências concernentes à Matemática. Em seguida, resumiremos o conteúdo matemático necessário para entender o desempenho(complexidade) de um Algoritmo de Ordenação e poder compará-los. Exemplificaremos alguns tipos de Algoritmos de Ordenação, seu funcionamento e a função matemática que mede seu desempenho. Por fim, iremos propor uma sequência de atividades que possam ser utilizadas no processo de ensino aprendizagem com alunos do Ensino Médio, salientando que o entendimento e desenvolvimento das mesmas devem se adequar a essa etapa do ensino de acordo com o ano, vivências de conteúdos, e período de maturação dos estudantes.

Palavras-chave: Parâmetros Curriculares, Competências, Habilidades, Algoritmos de Ordenação.

Abstract

Based on the new National Curriculum Common Base (BNCC), especially in skills related to algorithms, we will make a didactic approach to activities that can be worked in the teaching-learning process. Despite the skills and abilities required to apply algorithms, we will specifically apply sort algorithms without, however, observing or modeling them. We will analyze the National Curriculum Parameters: 5th to 8th grade, High School, BNCC (Common Basic National Curriculum) and also Curriculum Parameters of Basic Education of the State of Pernambuco; as the development of math-related skills and competencies. After that, we resume the mathematical content needed to understand the performance (complexity) of a Sorting Algorithm and comparable power. We also provide examples of various types of Sorting Algorithms, their functioning, and mathematical function that measure their performance. Finally, we propose a series of activities that can be used in the teaching-learning process with High School students, noting that the understanding and development of these should be appropriate to the process of teaching and learning according to the year, content experiences, and maturation period of the students.

Keywords: Curriculum Parameters, Competencies, Abilities, Sorting Algorithms.

Lista de ilustrações

Lista de tabelas

Sumário

Introdução

Segundo as Diretrizes Curriculares Nacionais da Educação Básica, (BRASIL, 2013)

> A UNESCO, em documento de 2007, ao entender que a qualidade da educação é também uma questão de direitos humanos,[...] advoga que a educação de qualidade, como um direito fundamental, deve ser antes de tudo relevante.[...]. A relevância reporta-se à promoção de aprendizagens significativas do ponto de vista das exigências sociais e de desenvolvimento pessoal.

De acordo com o mesmo documento (BRASIL, 2013, p. 109), "a escola, no desempenho de suas funções, deve construir e utilizar métodos, estratégias e recursos de ensino que melhor atendam às características cognitivas e culturais dos alunos". Diante dessa perspectiva, cientes que o uso de tecnologias é cada vez mais intenso na vida cotidiana e que a inserção crescente de pessoas jovens num mercado de trabalho altamente tecnológico requer mais preparo e formação específica nessas áreas, percebe-se quão grandes são os desafios da educação nos dias atuais, principalmente no que diz respeito à Matemática, que possui uma relação contígua com o desenvolvimento tecnológico.

Vários pesquisadores se debruçam na busca de novas metodologias que possam abarcar os conteúdos matemáticos de forma significativa e atraente, conectando-os com o mundo do trabalho e com o desenvolvimento científico. Propõe-se nessa pesquisa o tema Algoritmos de Ordenação, pois é possível instigar: a discussão sobre novas metodologias, a conexão com o desenvolvimento social, tecnológico e com o mundo do trabalho. Requer também a busca de resposta a uma situação-problema, de modo que a Matemática seja tratada com a formalidade requerida.

O estudo e a análise de Algoritmos de Ordenação no Ensino Médio podem ser de grande valia no processo de ensino e aprendizagem de Matemática, pois além de se apresentar como algo novo e atual, requer um trato matemático singular na resolução dos problemas propostos e favorece o desenvolvimento de habilidades e competências sugeridas na BNCC.

O principal objetivo deste trabalho é conhecer e analisar, à luz da BNCC, alguns Algoritmos de Ordenação utilizados em programação computacional, a saber, *Bubble Sort*, *Selection Sort* e *Quick Sort*, comparando suas eficiências por meio de funções Matemáticas. Objetivou-se ainda a construção de uma sequência didática que possa subsidiar o processo de ensino-aprendizagem no Ensino Médio. Há aqui um contraponto à pesquisa de Ribeiro (RIBEIRO, 2018, p. 27), que diz: "Propõe-se aqui o uso despretensioso da prática de programação no ambiente escolar como alternativa para despertar no aluno o interesse pela matemática[...]", expondo-se apenas a forma que um Algoritmo de Ordenação funciona, a análise de seu desempenho e a maneira que essa abordagem pode ser utilizada no processo de aprendizagem com estudantes do Ensino Médio. A escolha pelo Ensino Médio é justificável, pois supõe-se que nessa etapa da educação o estudante já está familiarizado com alguns conteúdos associados ao tema e tenha a maturidade suficiente para compreensão dos conceitos e procedimentos que serão abordados.

No primeiro capítulo realiza-se uma análise dos documentos: PCN-5ª a 8ª séries, PCNEM, PCNEM$_+$, Parâmetros para e Educação Básica do Estado de Pernambuco e BNCC, enfatizando competências e habilidades que possuem relação com o tema. Tais documentos são atualmente utilizados pelas redes e/ou sistemas de ensino na organização de suas propostas pedagógicas. A inclusão dos PCN-5ª a 8ª séries foi proposital, com o intuito de averiguar o que se propõe sobre o tema ao estudante antes da sua chegada no Ensino Médio.

No segundo, realizou-se um estudo da Matemática utilizada na análise dos Algoritmos de Ordenação, os conceitos utilizados e as funções envolvidas. Apesar de não fazer parte da proposta um aprofundamento desses conteúdos matemáticos, se faz necessário sua abordagem para que realize-se uma análise do algoritmo de maneira correta e formal. Também vale destacar que os conteúdos matemáticos sobre somatórios e propriedades normalmente não são abordados no Ensino Médio e podem ser passíveis de leitura e reflexão do professor do Ensino Médio mas o mesmo não se aplica se, supostamente, o leitor interessado for estudante do Ensino Médio. Em contrapartida, sugere-se que pode ser desnecessária uma leitura dos conteúdos de funções presentes neste capítulo por um professor do Ensino Médio contudo, caso um estudante dessa etapa se interesse pela leitura deste trabalho, se faz obrigatória tal leitura e reflexão.

No terceiro capítulo há uma apresentação dos Algoritmos de Ordenação já sugeridos, sua estrutura e a função Matemática que descreve sua eficiência (complexidade). Há ainda uso de fluxogramas para ilustrar o funcionamento de algoritmos, haja vista que se trata de uma habilidade sugerida pela BNCC. Explora-se exemplos e representações de algoritmos de modo que fique claro qual raciocínio dedutivo é utilizado na ordenação dos elementos de um conjunto finito e que função melhor representa tal situação sempre situando tais exemplos com as competências e habilidades requeridas no ensino-aprendizagem de Matemática.

Por fim, no último capítulo, propõe-se uma sequência didática que possa servir de parâmetro para o trabalho do professor do Ensino Médio. Nela consta desde atividades básicas de preenchimento de fluxogramas, leitura e produção de algoritmos, ordenação com número reduzido de elementos tendo como comando os Algoritmos de Ordenação já citados, bem como algumas atividades que avaliem a complexidade por meio de funções. A sequência pode ser utilizada pelo professor em todas as etapas do Ensino Médio, sendo que o mesmo deve avaliar se o estudante já possui condições de compreender a matemática utilizada na execução de alguma atividade.

Espera-se que a pesquisa aqui apresentada, de alguma forma possa auxiliar professores e, consequentemente, estudantes do Ensino Médio na aprendizagem dos Algoritmos de Ordenação de forma simples e inteligível, acrescentando mais uma ferramenta para o ensino de matemática. Também não se supõe que a abordagem do tema seja encerrado com o enfoque dado neste trabalho e sugere-se que outros pesquisadores possam acrescentar mais subsídios educacionais que possam ser utilizados como material de apoio ao aprendizado no Ensino Básico.

1 Análise dos Parâmetros Curriculares

Apresenta-se aqui uma breve análise de alguns documentos oficiais referentes ao Currículo do Ensino Básico Brasileiro sem se ter a pretensão de um aprofundamento ou estagnação da abordagem do tema nos documentos analisados. Busca-se mostrar uma possível evolução conceitual e curricular do tema Algoritmos de Ordenação em tais documentos e vislumbrar uma relativa importância de tal conteúdo no ensino-aprendizagem de Matemática no Ensino Médio. Apesar das recentes mudanças educacionais trazidas pela BNCC, a discussão acerca de uma aprendizagem mais significativa de matemática não é algo novo. Os PCN-5ª a 8ª séries, publicados em 1998, que corresponde atualmente a 6º ao 9º ano do Ensino Fundamental, já trazem tal problemática quando dizem:

> Situação-problema é o ponto de partida da atividade matemática e não a definição. No processo de ensino e aprendizagem, conceitos, ideias e métodos matemáticos devem ser abordados mediante a exploração de problemas, ou seja, de situações em que os alunos precisem desenvolver algum tipo de estratégia para resolvê-las. (BRASIL, 1998, p. 40)

O referido documento expõe que pela exploração de situações-problemas o aluno reconhecerá diferentes funções da Álgebra e representará problemas por meio de equações ou inequações. Acrescenta ainda sobre o exercício da indução e da dedução em Matemática, como algo revestido de importância no desenvolvimento da capacidade de resolver problemas, de formular e testar hipóteses, de induzir, de generalizar e de inferir dentro de determinada lógica, o que assegura um papel de relevo ao aprendizado dessa ciência em todos os níveis de ensino. Ao relacionar ideias Matemáticas entre si, o aluno pode reconhecer princípios gerais, como proporcionalidade, igualdade, composição, decomposição, inclusão e perceber que processos como o estabelecimento de analogias, indução e dedução. Os PCN-5ª a 8ª séries indicam como um dos objetivos do Ensino Fundamental que o aluno seja capaz de “questionar a realidade formulando-se problemas e tratando de resolvê-los, utilizando para isso o pensamento lógico, a criatividade, a intuição, a capacidade de análise crítica, selecionando procedimentos e verificando sua adequação”. (BRASIL, 1998, p. 8)

No que diz respeito ao ensino, “a prática mais frequente consiste em ensinar um conceito, procedimento ou técnica e depois apresentar um problema para avaliar se os alunos são capazes de empregar o que lhes foi ensinado (BRASIL, 1998, p. 40)”. Apesar dos PCN-5ª a 8ª séries abordarem que o uso da tecnologia da comunicação é um bom recurso no favorecimento à aprendizagem e de ressaltar a importância do computador e softwares nesse processo, ainda não se reporta ao uso de algoritmos no binômio ensino-aprendizagem, ou sequer cita sua existência. Em (BRASIL, 1998) afirma-se que “muitos conteúdos importantes serão descartados por serem julgados, sem uma análise adequada, que não são de interesse para os alunos porque não fazem parte de sua realidade ou não têm uma aplicação prática imediata". Como algoritmos podem ser representados por meio de fluxogramas, ressaltamos que os PCN-5ª a 8ª séries já consideram: "a visualização e a leitura de informações gráficas em Matemática são aspectos importantes, pois auxiliam a compreensão de conceitos e no desenvolvimento de capacidades de expressão gráficas (BRASIL, 1998, p. 46)". Algoritmo é um procedimento ou conjunto destes, vale ressaltar que os PCN-5ª a 8ª séries referem-se aos procedimentos como conteúdos em si, que possibilitam capacidades relacionadas ao saber fazer, sendo assim, os algoritmos enquadram-se em tal categoria de conteúdo. Enfim, mesmo com muitos indícios presentes nos PCN-5ª a 8ª séries sobre a importância do trabalho com novos métodos, conteúdos significativos e inovadores, bem como a exploração de situações-problemas que estimulem competências e habilidades específicas, não há explicitamente o estudo de Algoritmos de Ordenação como ferramenta de ensino-aprendizagem.

Observa-se nos PCNEM, na parte específica de Matemática, que há o tratamento dessa disciplina como uma ciência,

> com suas características estruturais específicas na qual o aluno percebe que as definições, demonstrações e encadeamentos conceituais e lógicos têm a função de construir novos conceitos a partir de outros e que servem para validar intuições e dar sentido às técnicas aplicadas. (BRASIL, 2000, pp. 40-41)

Também é no Ensino Médio que se propõe ao aluno uma ampliação dos conhecimentos adquiridos no Ensino Fundamental e uma elevação no nível de abstração de conceitos, "[...] deve ser mais do que memorizar resultados dessa ciência e que a aquisição do conhecimento matemático deve estar vinculada ao domínio de um saber fazer Matemática e de um saber pensar matemático (BRASIL, 2000, p. 41)". Outro viés explicitado nos PCNEM é a garantia de que o estudante seja capaz de lidar com o conceito de Funções em situações diversas, ser incentivado a buscar a solução, ajustando seus conhecimentos para construir um modelo para interpretação e investigação em Matemática. Também se faz importante que o aluno estenda e aprofunde seus conhecimentos sobre Números e Álgebra de forma conectada com outros conteúdos, pois estão intrinsecamente relacionados com a resolução de problemas. Em resumo, pode-se destacar nos PCNEM algumas habilidades e competências de base para desenvolver o ensino e a aprendizagem em Matemática, que são: as de representação e comunicação e, as de investigação e compreensão, que auxiliam o direcionamento com o tema aqui proposto.

> Os alunos, confrontados com situações-problema, novas, mas compatíveis com os instrumentos que já possuem ou que possam adquirir no processo, aprendem a desenvolver estratégia de enfrentamento, planejando etapas, estabelecendo relações, verificando regularidades, fazendo uso dos próprios erros cometidos para buscar novas alternativas; adquirem espírito de pesquisa, aprendendo a consultar, a experimentar, a organizar dados, a sistematizar resultados, a validar soluções; desenvolvem sua capacidade de raciocínio, adquirem auto-confiança e sentido de responsabilidade; e, finalmente, ampliam sua autonomia e capacidade de comunicação e de argumentação. (BRASIL, 2000, p. 52)

É possível continuar elencando conteúdos, procedimentos, habilidades e competências com as quais se promove uma aprendizagem significativa de Matemática, contudo não é proposta dessa pesquisa exaurir o tema, nem a discussão em torno dele. É perceptível a ausência de uma proposta explicita com o conteúdo Algoritmos e, em especial, os Algoritmos de Ordenação. Possivelmente pelo contexto histórico em que o documento PCNEM foi formulado ou ainda, pela falta de perspectiva dos pesquisadores sobre a viabilidade do trabalho com o tema.

Além dos PCNEM, foram publicados em 2002 os PCNEM_+ que teve o intuito de complementar as propostas de ensino e aprendizagem pautados em novos métodos e reflexões acerca dos conteúdos. Nesse documento há uma proposta de maior articulação entre as disciplinas curriculares de áreas afins, bem como a aproximação maior entre conteúdos e realidade. Os PCNEM_+ enfatizam as competências já elencadas nos PCNEM: representação e comunicação, investigação e compreensão e contextualização das ciências no âmbito sócio-cultural, reforçando a necessidade de compreensão destas, o modo como devem ser exploradas no meio escolar e indicando os tipos de conteúdos relacionados com cada competência. No que diz respeito à competência de representação e comunicação podemos destacar:

> Traduzir uma situação dada em determinada linguagem para outra; por exemplo, transformar situações dadas em linguagem discursiva em esquemas, tabelas, gráficos, desenhos, fórmulas ou equações matemáticas e vice-versa, assim como transformar as linguagens mais específicas umas nas outras, como tabelas em gráficos ou equações. (BRASIL, 2002, p. 115)

Vê-se claramente uma relação de tal competência com o estudo de Algoritmos de Ordenação, pois se faz necessário traduzir a linguagem do algoritmo para a linguagem matemática e para tanto se podem usar esquemas como fluxogramas. Ainda com respeito à competência de representação e comunicação faz-se destaque:

> Produzir textos analíticos para discutir, sintetizar e sistematizar formas de pensar, fazendo uso, sempre que necessário, da linguagem matemática. Redigir resumos, justificar raciocínios, propor situações-problema, sistematizar as ideias principais sobre dado tema matemático com exemplos e comentários próprios. (BRASIL, 2002, p. 115)

Na análise de Algoritmos de Ordenação se pretende que o estudante possa escrever a resposta a uma situação-problema justificando matematicamente seu raciocínio e utilizando, se necessário, diferentes formas de registro. Quanto à competência de investigação e compreensão, é relevante que o aluno possa identificar regularidades em situações semelhantes para estabelecer regras, algoritmos e propriedades, mas o trabalho com Algoritmos de Ordenação também requer que o estudante utilize “diferentes formas de quantificar dados numéricos para decidir se a resolução de um problema requer cálculo exato, aproximado, probabilístico ou análise de médias (BRASIL, 2002, p. 116)”. Já a competência de contextualização sócio-cultural pode ser relacionada com o ensino e aprendizagem de Algoritmos de Ordenação quando associa ao tema ciência e tecnologia na história, visto que

> A exigência de rapidez e complexidade dos cálculos fez com que a Matemática se desenvolvesse e, por outro lado, as pesquisas e avanços teóricos da Matemática e demais ciências permitiram o aperfeiçoamento de máquinas como o computador, que vêm tornando os cálculos cada vez mais rápidos. (BRASIL, 2002, p. 118)

É a criação de algoritmos que permitem que a rapidez desses cálculos seja obtida com êxito, mas é a Matemática, por meio da lógica usada na programação, que proporciona a real possibilidade para ocorrer tal sucesso. Os PCN_+ trazem um avanço na maneira de pensar e estruturas o processo de ensino-aprendizagem e propõe a seleção de Temas Estruturadores quando se quer almejar as competências já citadas. No entanto, se devem fazer alguns recortes utilizando alguns critérios de modo que os conteúdos ou temas devem: favorecer o desenvolvimento das competências elencadas, permitir uma articulação lógica entre diferentes ideias e conceitos, evitar detalhamentos ou nomenclaturas excessivas evitando um modelo curricular enfileirado como ocorre comumente. Entre esses Temas Estruturadores destacam-se Álgebra, Números e Funções, Análise de dados e Contagem. O destaque é pertinente à medida que o estudo de Algoritmos de Ordenação se pauta sobre tais temas e segundo os $PNCEM_+$ (BRASIL, 2002, p. 127),

> Esses conteúdos devem ter maior espaço e empenho de trabalho no Ensino Médio, mantendo de perto a perspectiva da resolução de problemas aplicados para se evitar a teorização excessiva e estéril.[...] Contribui também para a compreensão e o uso de representações gráficas, identificação de regularidades, interpretação e uso de modelos matemáticos e conhecimento de formas específicas de raciocinar em Matemática.

Dando continuidade ao que se propõe, ou seja, averiguar nos documentos oficiais algo que possa aludir o tema Algoritmos de Ordenação. Iremos realizar uma breve análise nos Parâmetros Curriculares da Educação Básica do Estado de Pernambuco, publicado em 2012. Nele há tópicos semelhantes aos encontrados nos Parâmetros Curriculares Nacionais e BNCC, como resolução de problemas e modelagem matemática, no entanto, há uma mudança na nomenclatura quando se refere à competência. Tais competências são chamadas de Objetivos nos PCN-5ª a 8ª séries, Competências e Habilidades nos PCNEM e $PCNEM_+$. Já nos Parâmetros Curriculares da Educação Básica do Estado de Pernambuco o nome dado, ao objetivo que se pretenda que o estudante alcance, é Expectativa de Aprendizagem.

> As expectativas de aprendizagem explicitam aquele mínimo que o estudante deve aprender para desenvolver as competências básicas na disciplina. Em outras palavras, elas descrevem o "piso" de aprendizagens, e não o "teto". Dependendo das condições de cada sala de aula, elas podem ser ampliadas e/ou aprofundadas. (PERNAMBUCO, 2012, p. 13)

Apesar dos Parâmetros Curriculares da Educação Básica do Estado de Pernambuco terem se baseado na Base Curricular Comum para as Redes Públicas de Ensino de Pernambuco (BCC-PE, Secretaria de Educação de Pernambuco, 2008), não foi observado aqui o texto da mesma, ficando a análise restrita aos Parâmetros Curriculares. Neste último, a estratégia de resolução de uma situação-problema é tida como privilegiada, onde problemas abertos devem ser prioritários em detrimento dos fechados. Também ressalta a modelagem matemática como algo com estreita conexão com a resolução de problemas.

> De fato, quando a modelagem matemática propõe uma situação-problema ligada ao "mundo real", com sua inerente complexidade, o estudante é chamado a mobilizar um leque variado de competências: selecionar variáveis que serão relevantes para o modelo a construir; problematizar, ou seja, formular um problema teórico, na linguagem do campo matemático envolvido; formular hipóteses explicativas do fenômeno em causa; recorrer ao conhecimento matemático acumulado para a resolução do problema formulado, o que, muitas vezes, requer um esforço de simplificação, pelo fato de que o modelo originalmente pensado pode revelar-se matematicamente muito complexo; validar, isto é, confrontar as conclusões teóricas com os dados empíricos existentes, o que, quase sempre, leva à necessidade de modificação do modelo, que é essencial para revelar o aspecto dinâmico da construção do conhecimento. (PERNAMBUCO, 2012, p. 30)

Mesmo com a mudança de nomenclatura, as competências, associadas às habilidades, estão intrinsecamente ligadas às Expectativas de Aprendizagem, como consta no próprio documento: "o trabalho com os saberes, no entanto, deve ser orientado para as competências que se deseja que o estudante construa, o que nos leva à necessidade de estabelecer as expectativas de aprendizagem (PERNAMBUCO, 2012, p. 21)". Porém, o documento levanta a bandeira que por meio de aquisição de "saberes" as competências devem ser construídas e não o contrário. Concordando, no entanto, com a concepção de que é por meio de uma situação-problema que a aprendizagem de um novo conceito ocorre. Dentre as expectativas de aprendizagem observa-se, no quadro abaixo, algumas que possuem forte relação com o ensino e aprendizagem de Algoritmos de Ordenação. Apesar do foco desta pesquisa ser o Ensino Médio, incluímos nessa tabela o 9º ano com o intuito de mostrar que algumas competências e habilidades ou Expectativas de Aprendizagem já se fazem presentes no Ensino Fundamental e deverão ser consolidados no Ensino Médio.

Tabela 1 – Expectativas de Aprendizagem: Parâmetros Curriculares do Estado de Pernambuco

Expectativa de Aprendizagem	9ºano	10ºano	11ºano	12ºano
Comparar e ordenar números reais	X	-	-	-
Reconhecer função como modelo matemático para o estudo das variações entre grandezas do mundo natural ou social	-	X	X	X
Diferenciar Modelo de crescimento/decrescimento de função exponencial em relação às funções lineares e quadráticas	-	X	X	-

Fonte: Produzida pelo autor

Apesar da expectativa envolvendo comparação e ordenação não aparecer no Ensino Médio, salientamos que o estudo com Algoritmos de Ordenação aqui proposto é voltado principalmente à sua relação com a função que mede sua eficiência, e tal função está relacionada com a forma que a contagem de ações de ordenação é efetuada. Assim, tal expectativa já se encontra implícita no processo sem que haja necessidade de inclusão prioritária da mesma no Ensino Médio. Observamos também a falta de referência à Função Logarítmica nas Expectativas de Aprendizagem do Ensino Médio, contudo a mesma aparece na BNCC como será visto adiante.

Iniciando a breve análise da BNCC, assim como foram observadas algumas características dos PCN-5ª a 8ª séries se faz importante uma abordagem, na parte específica de Matemática, do Ensino Fundamental na BNCC. Sabe-se que, de acordo com os documentos curriculares oficiais até aqui analisados, o estudante do Ensino Médio é orientado a aprofundar e ampliar os conceitos e procedimentos vivenciados no Ensino Fundamental. O desenvolvimento da aprendizagem é proposto, nesse documento, pelo desenvolvimento de competências e habilidades de forma semelhante aos PCNEM. De acordo com a BNCC (BRASIL, 2018, p. 8),

> Competência é definida como a mobilização de conhecimentos (conceitos e procedimentos), habilidades (práticas, cognitivas e socioemocionais), atitudes e valores para resolver demandas complexas da vida cotidiana, do pleno exercício da cidadania e do mundo do trabalho.

Em (BRASIL, 2018) são explicitadas Competências Gerais da Educação Básica e Competências Específicas de Cada Área de Conhecimento dentro das quais são indicadas as habilidades requeridas. Na BNCC do Ensino Fundamental os conteúdos são separados em unidades temáticas do 1º ao 9º ano e organizadas da seguinte maneira: Números, Álgebra, Geometria, Grandezas e Medidas, Probabilidade e Estatística. A BNCC explicita a importância de se relacionar, sempre que possível, as unidades temáticas supra citadas no processo de ensino-aprendizagem. No entanto, a organização curricular se dá de uma forma diferente da que aparece no Ensino Médio como veremos adiante. No Ensino Fundamental as habilidades são relacionadas diretamente, no arranjo curricular, às unidades temáticas e aos objetos de conhecimento. No que diz respeito ao tema Algoritmos, o documento defende que este tem relação bastante próxima do pensamento computacional, que por sua vez tem bastante importância, pois propicia uma variedade de representações e uso de diferentes linguagens. Segundo o documento, "A linguagem algorítmica tem pontos em comum com a linguagem algébrica, sobretudo em relação ao conceito de variável (BRASIL, 2018, p. 271)".

Podemos destacar dentre as competências, que se referem ao Ensino Fundamental na BNCC, algumas que possuem relação com o tema algoritmos:

> -Compreender as relações entre conceitos e procedimentos dos diferentes campos da Matemática (Aritmética, Álgebra, Geometria, Estatística e Probabilidade) e de outras áreas do conhecimento, sentindo segurança quanto à própria capacidade de construir e aplicar conhecimentos matemáticos, desenvolvendo a autoestima e a perseverança na busca de soluções. -Enfrentar situações-problema em múltiplos contextos, incluindo-se situações imaginadas, não diretamente relacionadas com o aspecto prático-utilitário, expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens (gráficos, tabelas, esquemas, além de texto escrito na língua materna e outras linguagens para descrever algoritmos, como fluxogramas, e dados). (BRASIL, 2018, p. 267)

Quanto às habilidades que interligam ao tema algoritmos, encontram-se principalmente nos anos finais dos Ensino fundamental, sendo tratados nos anos iniciais com significado mais voltado para conceitos e procedimentos matemáticos, em detrimento do aspecto computacional. Dentre as habilidades relacionadas ao tema no Ensino Fundamental, destacamos:

> -Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas.
> -Descrever, por escrito e por meio de um fluxograma, um algoritmo para a construção de um triângulo qualquer, conhecidas as medidas dos três lados.
> -Descrever, por escrito e por meio de um fluxograma, um algoritmo para a construção de um polígono regular (como quadrado e triângulo equilátero), conhecida a medida de seu lado.
> -Identificar a regularidade de uma sequência numérica ou figural não recursiva e construir um algoritmo por meio de um fluxograma que permita indicar os números ou as figuras seguintes. (BRASIL, 2018)

Percebe-se que se apresentou, na BNCC do Ensino Fundamental, uma proposta buscando um uso intencional de fluxogramas em diversas temáticas, bem como associar com o conceito de algoritmo voltado mais para Matemática, mas a relação com o conceito computacional ainda não foi foco nesta seção do documento.

Não é propósito deste trabalho aprofundar o processo de construção e organização da BNCC, a pesquisa restringe-se principalmente na análise das competências e habilidades específicas da Área de Matemática no Ensino Médio. Assim, é importante apontar que na BNCC, no que diz respeito à influência das Tecnologias digitais e Computação, há indicação de uma competência destacando: “pensamento computacional: envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos (BRASIL, 2018, p. 474)”. Apesar da BNCC fazer uso de itinerários formativos, não faz parte do presente estudo adentrar no mérito da questão nem tampouco o foco de tais itinerários interfere na análise aqui proposta.

Seguindo com a verificação, na parte específica de Matemática, a BNCC ressalta que os estudantes “ao chegarem aos anos finais, eles possam ser estimulados a desenvolver o pensamento computacional, por meio da interpretação e da elaboração de algoritmos, incluindo aqueles que podem ser representados por fluxogramas”. (BRASIL, 2018, p. 528). Também devem desenvolver habilidades relativas à investigação, construção de modelos e resolução de problemas e serem estimulados de forma a favorecer o desenvolvimento de competências que envolvam raciocinar, representar, comunicar e argumentar. Algumas competências específicas da área de matemática podem receber destaque:

> -Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente.
> -Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando estratégias e recursos, como

> observação de padrões, experimentações e diferentes tecnologias, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas. (BRASIL, 2018, p. 531)

Os Algoritmos de Ordenação se apresentam assim como um tema pertinente para um trabalho de ensino e aprendizagem que atendam as competências acima destacadas, pois as habilidades relacionadas a elas requerem procedimentos de identificação, leitura, interpretação, construção de modelos, resolução e elaboração de problemas, e análise de situações e dados ligados ao mundo real e em especial aos avanços tecnológicos. Dentre as habilidades relacionadas às competências específicas as que se relacionam mais intrinsecamente com Algoritmos de Ordenação são:

> -Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema.
> -Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática. (BRASIL, 2018, p. 538-539)

Vale salientar que existem outras competências e habilidades específicas na BNCC que também possuem relação com os Algoritmos de Ordenação e é importante lembrar que não é foco deste estudo a programação computacional, mas também não está nos propósitos uma avaliação minuciosa do campo conceitual em questão. Portanto, apenas as competências e habilidades que se fazem necessárias no direcionamento do estudo em tela.

Buscou-se realizar neste capítulo uma análise, seguindo uma ordem cronológica, dos Parâmetros Curriculares brasileiros: PCN-5ª a 8ª séries (1998), PCNEM (2000), PCNEM$_+$ (2002), Parâmetros para a educação Básica do Estado de Pernambuco (2012) e BNCC (2018). A intenção foi detectar as eventuais mudanças curriculares e de paradigmas na abordagem de conteúdos que se assemelhem ou relacionem com os Algoritmos de Ordenação. Pôde-se observar em nossa breve análise, dessas diretrizes curriculares para o âmbito educacional federal, estadual e municipal, que são várias as competências, habilidades e, no caso de Pernambuco, Expectativas de Aprendizagem, indicando a viabilidade com o tema aqui proposto sendo que apenas na BNCC (2018) o conteúdo está explicito. Averiguou-se que a atenção ao tema só foi dada recentemente e acredita-se que se faz necessário uma busca por estratégias didáticas que tornem o tema tangível a professores e alunos do Ensino Médio.

2 Matemática e Algoritmos de Ordenação

Neste capítulo será explicitado um resumo dos conceitos e resultados matemáticos necessários para uma análise segura dos Algoritmos de Ordenação, tanto no que diz respeito ao funcionamento, bem como sua complexidade. Abordar-se-ão as funções que estão relacionadas com tais algoritmos para serem utilizadas no capítulo subsequente, onde será realizado o estudo da complexidade desses algoritmos. Como já foi indicado na parte introdutória deste trabalho, o professor do Ensino Médio pode abster-se da leitura e reflexão do conteúdo referente às Funções Afim, Quadrática e Logarítmica, pois se trata de um conteúdo recorrente no Ensino Médio. No entanto, sugere-se que tal leitura e reflexão seja indicada para um Estudante do Ensino Médio que se interesse pelo trabalho aqui desenvolvido.

Apesar dos estudos de alguns algoritmos de ordenação requererem uma Matemática mais avançada para sua análise, ater-se aos conceitos e métodos mais acessíveis ao Ensino Médio não caracteriza prejuízo. Será utilizado como texto base o livro *Algoritmos Teoria e Prática*, cujo autor é Thomas H. Cormen[*et.al*] (CORMEN, 2002), para abordar os conteúdos de Somatórios e Séries. Já para os conteúdos Funções Afim, Quadrática e Logarítmica, o Livro texto utilizado é *Conexões com a matemática*, Ensino Médio, Org. Editora Moderna, 3.ed., São Paulo, 2016, vol.1. A última escolha se justifica, pois além do intuito da pesquisa ser voltada para o público do Ensino Médio, esse é um dos livros adotados pela Rede Estadual do Estado de Pernambuco sendo suficiente para subsidiar a análise e a comparação dos Algoritmos de Ordenação aqui analisados.

Segundo (CORMEN, 2002), “Quando um algoritmo contém uma construção de controle interativo (ou repetitivo)[...] seu tempo de execução pode ser expresso como a soma dos tempos gastos em cada execução[...]”. Daí a necessidade da inclusão do tema somatórios e propriedades, sem pretensão de nenhum aprofundamento desnecessário. Apesar de haver na análise de algoritmos que usam a técnica dividir e conquistar cálculos com recorrência, não será tratado esse conteúdo, visto que busca-se realizar tal análise apenas com a matemática tangível ao estudante do Ensino Médio. Acrescentar-se-á ainda, conceitos relacionados a Função afim, Função quadrática, Logaritmos e propriedades e Função logarítmica, pois serão utilizados na modelagem matemática e na comparação da complexidade dos Algoritmos de Ordenação estudados.

No capítulo seguinte serão abordados os desempenhos dos Algoritmos de Ordenação no melhor caso, no pior e no caso médio. Assim, vale adiantar que a Função Afim representa o melhor caso do algoritmo *Bubble Sort*. Já a Função Quadrática é utilizada na representação do pior caso e caso médio do *Bubble Sort*, em qualquer caso do *Selection Sort* e também no pior caso do *Quick Sort*. As propriedades dos Logaritmos e a Função Logarítmica são requeridas para conjecturar e representar o desempenho do algoritmo *Quick Sort*.

2.1 Somatórios e propriedades

Dada uma sequência de números $a_1, a_2, a_3, ...$, a soma finita $a_1 + a_2 + ... + a_n$ em que n é um número inteiro positivo, pode ser escrita em uma notação de somatório na forma

$$\sum_{j=1}^{n} a_j.$$

O valor de uma soma finita é sempre bem definido, e seus termos podem ser somados em qualquer ordem. Dada uma sequência $a_1, a_2, a_3, ...$ a soma infinita $a_1 + a_2 + ...$ pode ser escrita como

$$\sum_{k=1}^{\infty} a_k,$$

cuja convergência é regida pelo limite

$$\lim_{n \to +\infty} \sum_{k=1}^{n} a_k.$$

Se o limite existe, a série ***converge***, caso contrário ***diverge***.

Para qualquer número real c e quaisquer sequências finitas $a_1 + a_2 + ... + a_n = \sum_{k=1}^{n} a_k$ e $b_1 + b_2 + ... + b_n = \sum_{k=1}^{n} b_k$, valem as seguintes propriedades de **linearidade**

i) $\sum_{k=1}^{n}(a_k + b_k) = \sum_{k=1}^{n} a_k + \sum_{k=1}^{n} b_k.$

ii) Dado, $c \in \mathbb{R}$, tem-se que

$$\sum_{k=1}^{n}(c \cdot a_k + b_k) = c \cdot \sum_{k=1}^{n} a_k + \sum_{k=1}^{n} b_k.$$

Estas propriedades são igualmente válidas caso se tenha uma série convergente.

Exemplo 2.1. Sabendo-se que o somatório $\sum_{k=1}^{n} k = 1 + 2 + ... + n$, é uma ***Série Aritmética*** e tem o valor $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$. Deseja-se calcular o valor de $\sum_{k=1}^{n}(2k+1)$.

De fato, pode-se escrever, segundo as propriedades de linearidade

$$\sum_{k=1}^{n}(2k+1) = 2 \cdot \sum_{k=1}^{n} k + \sum_{k=1}^{n}(1),$$

que equivale a escrever:

$$\sum_{k=1}^{n}(2k+1) = 2\cdot\frac{n(n+1)}{2} + n = n(n+2).$$

Exemplo 2.2. Sabendo-se $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$ é **Soma dos Quadrados dos primeiros n inteiros positivos**, deseja-se calcular o valor de $\sum_{k=1}^{n}(2k^2+1)$.

De fato, novamente usando-se as propriedades de linearidade

$$\sum_{k=1}^{n}(k^2+1) = 2\cdot\sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} 1,$$

que é equivalente a:

$$\sum_{k=1}^{n}(2k^2+1) = 2\sum_{k=1}^{n} k^2 + n = \frac{n(n+1)(2n+1)}{3} + n = \frac{n\left[(n+1)(2n+1)+3\right]}{3}$$

Exemplo 2.3. $\sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2$ é a **Soma dos Cubos dos primeiros n inteiros positivos**. Deseja-se calcular o valor de $\sum_{k=1}^{n}(k+1)^3$.

De fato, pode-se escrever pelas propriedades de linearidade

$$\sum_{k=1}^{n}(k+1)^3 = 2^3+3^3+\ldots+n^3+(n+1)^3+1-1 = \sum_{k=1}^{n} k^3+(n+1)^3-1 = \left(\frac{n(n+1)}{2}\right)^2+(n+1)^3-1.$$

Exemplo 2.4. Para o número real $x \neq 1$, o somatório $\sum_{k=0}^{n} x^k = 1 + x^2 + x^3 + ... + x^n$ é uma ***Série Geométrica*** ou ***Série Exponencial*** e tem valor

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}.$$

No caso em que $x = 1$ temos:

$$\sum_{k=0}^{n} x^k = 1 + x^2 + x^3 + ... + x^n = 1 + \sum_{k=1}^{n} x^k = 1 + \sum_{k=1}^{n} 1^k = 1 + n.$$

Exemplo 2.5. Quando o somatório é infinito e $|x| < 1$, temos a série geométrica decrescente e

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}.$$

Não haverá um aprofundamento sobre as propriedades dos somatórios para ***Séries Aritméticas*** e ***Séries Geométricas***, pois não é necessário para o escopo do trabalho aqui proposto. Havendo apenas um direcionamento sobre séries que representam Progressões Aritméticas (PA) e Progressões Geométricas (PG) assim denominadas, bem como ensinadas no Ensino Médio. As séries aritméticas serão requeridas no cálculo do número de comparações dos Algoritmoas de Ordenação *Bubble Sort* e *Selection Sort*, já as séries geométricas têm relevância no cálculo do número de comparações realizado pelo *Quick Sort*.

2.2 Função Afim

Esta seção se reporta a uma breve abordagem da ***Função Afim*** haja vista que a análise e comparação da complexidade de algoritmos de Ordenação, como já foi dito no início deste capítulo, requer a comparação gráfica das funções envolvidas das quais faz parte a ***Função Afim*** que será utilizada na observação do melhor caso do algoritmo *Bubble Sort*.

De acordo com o livro texto aqui adotado: "Uma função $f : \mathbb{R} \to \mathbb{R}$ chama-se ***Função Afim*** quando existem números reais a e b tais que $f(x) = ax + b$ para todo $x \in \mathbb{R}$ (LEONARDO, 2016, p. 85)".

Exemplo 2.6. A função $g : \mathbb{R} \to \mathbb{R}$ tal que $g(x) = -\frac{1}{2}x + 5$ em que $a = -\frac{1}{2}$ e $b = 5$, é uma ***Função Afim***.

Casos particulares de Função Afim

Uma função $f : \mathbb{R} \to \mathbb{R}$ chama-se ***Função Constante*** quando existe um número real b tal que $f(x) = b$ para todo $x \in \mathbb{R}$.

Exemplo 2.7. Uma função $t : \mathbb{R} \to \mathbb{R}$ tal que $t(x) = -5$ em que $a = 0$ e $b = -5$, é uma ***Função Constante***.

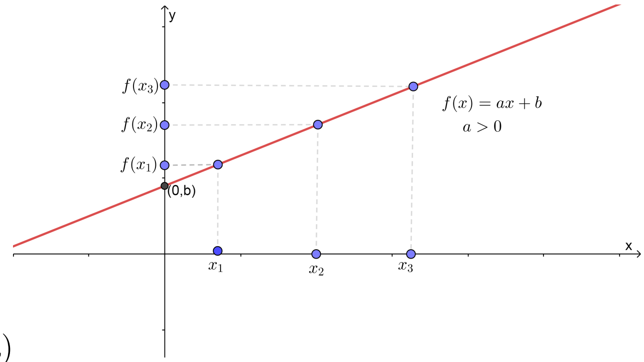
Uma função $f : \mathbb{R} \to \mathbb{R}$ chama-se ***Função Polinomial do 1º grau*** quando existem números reais a e b, com $a \neq 0$ tais que $f(x) = ax + b$ para todo $x \in \mathbb{R}$. Uma **Função Polinomial do 1º grau** que tem o coeficiente $b = 0$ recebe o nome de ***Função Linear***.

Exemplo 2.8. A função $h : \mathbb{R} \to \mathbb{R}$ tal que $h(x) = -7x$ em que $a = -7$ e $b = 0$, é uma ***Função Linear***.

Gráfico da Função Afim

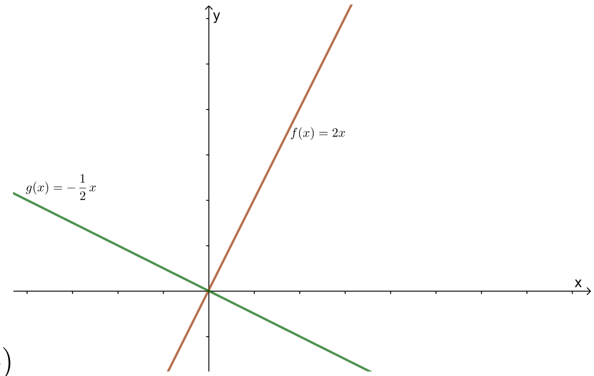
O gráfico de uma ***Função Afim*** é uma reta oblíqua aos eixos Ox e Oy.

Figura 1 – Gráficos de Funções Afins



(a)

Função Afim



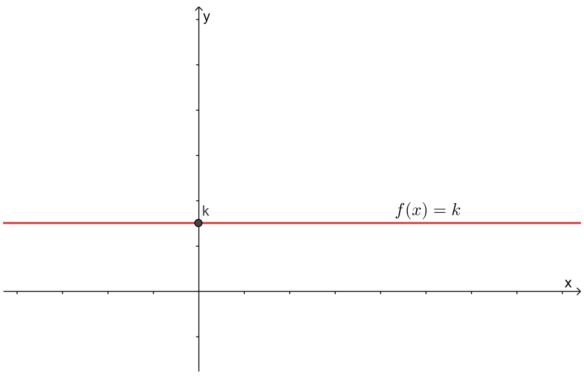
(b)

Função Linear

Fonte: Produzido pelo autor

No caso do gráfico de uma **Função Constante** temos uma reta paralela ao eixo O_x.

Figura 2 – Gráfico da Função Constante

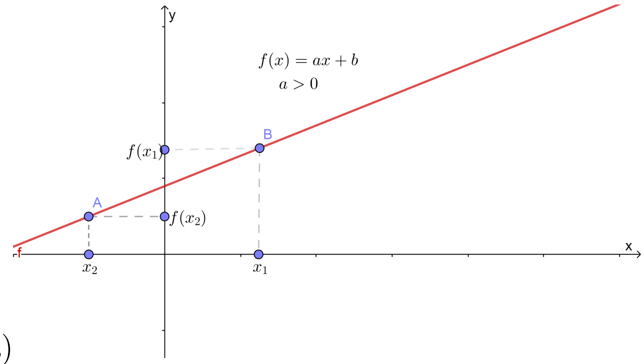


Fonte: Produzido pelo autor

Análise do gráfico da Função Afim

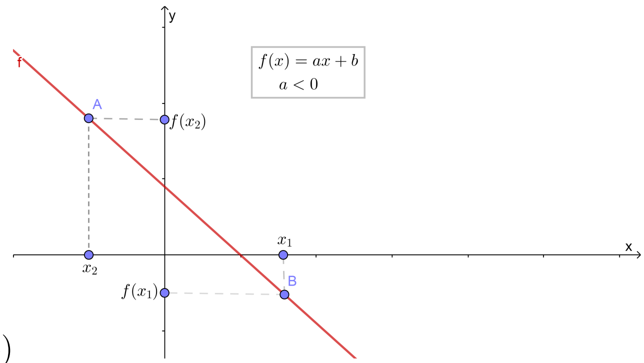
Uma ***Função Afim***, de lei $f(x) = ax + b$, pode ser crescente, decrescente ou constante, dependendo do valor do coeficiente a da função.

Figura 3 – Gráfico da Função Afim crescente e decrescente



(a)

Função crescente



(b)

Função decrescente

De um modo geral:

- Para uma Função Afim com coeficiente $a > 0$, temos:
  $\forall\ x_2 < x_1 \Rightarrow ax_2 < ax_1 \Rightarrow ax_2 + b < ax_1 + b$, ou seja, $f(x_2) < f(x_1)$ e $f(x)$ é ***crescente***.

- Para uma Função Afim com coeficiente $a < 0$ temos:
  $\forall\ x_2 < x_1 \Rightarrow ax_2 > ax_1 \Rightarrow ax_2 + b > ax_1 + b$, ou seja, $f(x_2) > f(x_1)$ e $f(x)$ é ***decrescente***.

2.3 Função Quadrática

"Uma função $f : \mathbb{R} \to \mathbb{R}$ chama-se ***Função Quadrática*** ou ***Função Polinomial do 2º grau*** quando existem números reais a, b e c, com $a \neq 0$, tais que $f(x) = ax^2 + bx + c$, para todo $x \in \mathbb{R}$ (LEONARDO, 2016, p. 108)."

Exemplo 2.9. Seja $f : \mathbb{R} \to \mathbb{R}$ uma função dada por $f(x) = 2x^2 + 3x - 15$, em que $a = 2$, $b = 3$ e $c = -15$.

Exemplo 2.10. Seja $g : \mathbb{R} \to \mathbb{R}$ uma função dada por $g(x) = -\frac{x^2}{4} + 5$, em que $a = -\frac{1}{4}$, $b = 0$ e $c = 5$.

Exemplo 2.11. Seja $h : \mathbb{R} \to \mathbb{R}$ uma função dada por $h(x) = -x + \sqrt{2}x^2$, em que $a = \sqrt{2}$, $b = -1$ e $c = 0$.

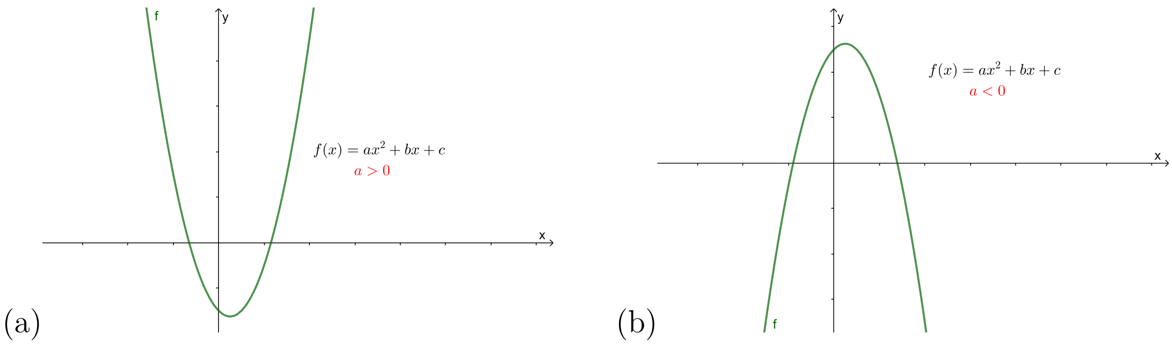
Exemplo 2.12. Seja $i : \mathbb{R} \to \mathbb{R}$ uma função dada por $i(x) = -\frac{3}{2}x^2$, em que $a = -\frac{3}{2}$, $b = 0$ e $c = 0$.

Gráfico da Função Quadrática

O gráfico de uma ***Função Quadrática*** é uma curva no formato de uma **parábola**.

Quando representam uma Função Quadrática, as **parábolas** podem ter a abertura (**concavidade**) voltada para cima ou para baixo. Na prática, observamos o sinal do coeficiente a da ***Função Quadrática*** dada por $f(x) = ax^2 + bx + c$ para determinar o sentido da concavidade da parábola como está ilustrado na figura seguinte.

Figura 4 – Gráficos de Funções Quadráticas



Fonte: Produzido pelo autor

- Se $a > 0$, então a parábola que representa a função f tem concavidade voltada para **cima**.
- Se $a < 0$, então a parábola que representa a função f tem concavidade voltada para **baixo**.

Elementos do Gráfico da Função Quadrática

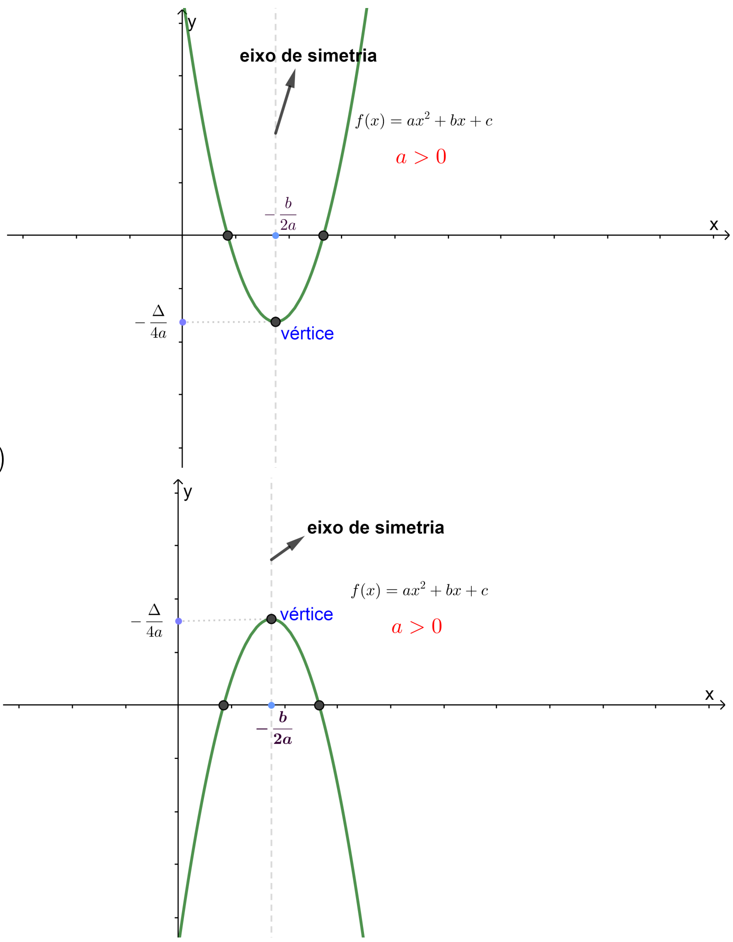
É importante identificar os elementos de uma **parábola** que representa uma ***Função Quadrática***, pois tais elementos são utilizados como parâmetros na análise de tal função.

Vértice da parábola

Na parábola que representa uma Função Quadrática, o ***vértice*** corresponde ao ponto da curva de menor ordenada (quando $a > 0$) ou de maior ordenada (quando $a < 0$), cujas coordenadas são representadas por $V(x_v, y_v)$. A reta perpendicular ao eixo O_x que passa pelo ***vértice*** da parábola, é denominada **eixo de simetria** e seus pontos são tais que $x = x_v$.

As coordenadas do ***vértice*** de uma **parábola** que representa uma Função Quadrática de lei $f(x) = ax^2 + bx + c$, com $a \neq 0$, como ilustrado na figura a seguir, são dadas por $x_v = \dfrac{-b}{2a}$ e $y_v = \dfrac{-\Delta}{4a}$. As demostrações correspondentes foram omitidas, pois não são foco deste trabalho.

Figura 5 – Gráfico indicando o vértice da parábola que representa uma Função Quadrática



(a)

(b)

Fonte: Produzido pelo autor

Intervalo de crescimento e decrescimento da função quadrática

Para analisar o intervalo de **crescimento** e **decrescimento** da ***Função Quadrática*** devemos observar a abscissa do vértice (x_v) como ilustrado na figura 5 observa-se as seguintes situações.

Quando $a > 0$ temos:
Se $x < x_v$, então a função é **decrescente**, caso contrário é **crescente**.

Demonstração 1. *Sejam x_1 e x_2 dois pontos quaisquer do eixo das abscissas tais que $x_1 < x_2 < x_v$, temos*

$$x_1 < \frac{-b}{2a} \quad e \quad x_2 < \frac{-b}{2a} \Longrightarrow x_1 + x_2 < \frac{-2b}{2a} = \frac{-b}{a}.$$

Daí,

$$x_1 + x_2 < \frac{-b}{a}.$$

Como $a > 0$, multiplicando os dois membros da desigualdade por a obtemos

$$a \cdot (x_1 + x_2) < -b.$$

Temos ainda que $(x_1 - x_2) < 0$, multiplicando os dois membros da última desigualdade por este fator obtém-se

$$a \cdot (x_1 + x_2) \cdot (x_1 - x_2) > -b \cdot (x_1 - x_2) \Longrightarrow (ax_1^2 - ax_2^2) > -bx_1 + bx_2 \Longrightarrow ax_1^2 + bx_1 > ax_2^2 + bx_2.$$

Adicionando um número real c a ambos os membros ficamos com

$$ax_1^2 + bx_1 + c > ax_2^2 + bx_2 + c \Longrightarrow f(x_1) > f(x_2)$$

Portanto, quando $a > 0$, temos para todo $x < x_v$, $f(x_1) > f(x_2)$ e a função é decrescente. Analogamente, para todo $x > x_v$, $f(x_1) < f(x_2)$ e a função é crescente.

Quando $a < 0$ temos:
Se $x < x_v$, então a função é **crescente**, caso contrário é **decrescente**.
A prova é análoga à Demonstração 1.

Independente de $a > 0$ ou $a < 0$ temos:
Se $x = x_v$, então a função nem é crescente e nem decrescente.

2.4 Função Logarítmica

Como um dos focos nesta pesquisa é a análise da complexidade dos Algoritmos de Ordenação, não consta aqui todo conteúdo relacionado aos logaritmos, mas apenas o que diz respeito a algumas definições e propriedades operatórias, bem como ***Função Logarítmica*** e sua análise gráfica. Antes, faremos uma breve abordagem sobre potências e Função Exponencial com o intuito de basilar melhor a abordagem com a Função Logaritma. Para tanto, abrimos espaço para uma bibliografia complementar do livro: *A Matemática do Ensino Médio*, Vol. 1, SBM, 1997. Cabe observar que em (LEONARDO, 2016) o conjunto dos reais positivos é indicado por $\mathbb{R}_+^*$, enquanto em (LIMA et al., 1997) é indicado por $\mathbb{R}^+$.

Função Exponencial

Potências de expoente racional

Para todo $n \in \mathbb{N}$, a potência a^n, de base a e expoente n é definida como o produto de n fatores iguais a a, ou seja, $a^n = \underbrace{a \cdot a \cdot \ldots \cdot a}_{n \quad fatores}$.

Para $n = 1$, temos por definição que $a^1 = a$, fazendo indução em a^n obtemos $a^{n+1} = a \cdot a^n$ e podemos afirmar que a^n é valido para todo $n \geq 1$.

Temos, para quaisquer m, $n \in \mathbb{N}$,

$$a^m \cdot a^n = \underbrace{a \cdot a \cdot \ldots \cdot a}_{n \quad fatores} \cdot \underbrace{a \cdot a \cdot \ldots \cdot a}_{m \quad fatores} = \underbrace{a \cdot a \cdot \ldots \cdot a}_{n+m \quad fatores} = a^{m+n}.$$

De posse do último resultado, para quaisquer $m_1, m_2, \ldots, m_k$, vale a igualdade

$$a^{m_1} \cdot a^{m_2} \cdot \ldots \cdot a^{m_k} = a^{m_1+m_2+\ldots+m_k},$$

e de modo particular, sendo $m_1 = \ldots = m_k = m$, temos $(a^m)^k = a^{m \cdot k}$.

Sendo $a > 1$, como $a^n > 0$, podemos escrever $a^{n+1} > a^n$. Portanto,

$$a > 1 \Rightarrow 1 < a < a^2 < \ldots < a^n < a^{n+1} \ldots \quad .$$

De modo análogo,

$$0 < a < 1 \Rightarrow 1 > a > a^2 > \ldots > a^n > a^{n+1} > \ldots \quad .$$

Portanto a sequência cujo o n-ésimo termo é a^n é:

- crescente quando $a > 1$;
- decrescente se $0 < a < 1$;
- constante quando $a = 1$.

No caso de n ser inteiro, o significado da potência a^n deve satisfazer a regra fundamental $a^m \cdot a^n = a^{m+n}$.

Vejamos para n=0, neste caso a igualdade $a^0 \cdot a^1 = a^{0+1}$ deve ser válida, ou seja, $a^0 \cdot a = a$ e portanto $a^0 = 1$.

O significado de a^{-n} pode ser obtido observando-se que, para qualquer $n \in \mathbb{N}$, $-n$ é negativo e devemos ter $a^{-n} \cdot a^n = a^{-n+n} = a^0 = 1$, ou seja, $a^{-n} = \dfrac{1}{a^n}$.

A função $f : \mathbb{Z} \to \mathbb{R}^+$, dada por $f(n) = a^n$, $n \in \mathbb{Z}$, além de cumprir a igualdade fundamental é ainda crescente quando $a > 1$ e decrescente se $0 < a < 1$. Em particular,

- para $a > 1$ e $n \in \mathbb{N}$, tem-se $a^{-n} < 1 < a^n$;
- para $0 < a < 1$ e $n \in \mathbb{N}$, tem-se $a^n < 1 < a^{-n}$.

De $a^m \cdot a^n = a^{m+n}$, segue-se que

$$(a^m)^n = \underbrace{a^m \cdot a^m \cdot \ldots \cdot a^m}_{n \quad fatores} = \underbrace{\underbrace{a \cdot a \cdot \ldots \cdot a}_{m \quad fatores} \cdot \underbrace{a \cdot a \cdot \ldots \cdot a}_{m \quad fatores} \cdot \ldots \cdot \underbrace{a \cdot a \cdot \ldots \cdot a}_{m \quad fatores}}_{n \cdot m \quad fatores} = a^{m \cdot n}$$

com m, $n \in \mathbb{Z}$.

Agora, vejamos o sentido que pode ser dado à potência a^r quando $r = \frac{m}{n}$, com $m \in \mathbb{Z}$ e $n \in \mathbb{N}$. Mantendo válida a igualdade $a^r \cdot a^s = a^{r+s}$, temos

$$(a^r)^n = a^r \cdot a^r \cdot \ldots \cdot a^r = a^{r+r+\ldots+r} = a^{r \cdot n} = a^m.$$

Portanto a^r é o número real positivo cuja n-ésima potência é igual a a^m. Por definição de raiz, este número é $\sqrt[n]{a^m}$. Assim, a única maneira de definir a potência a^r, com $r = \frac{m}{n}$, $m \in \mathbb{Z}$ e $n \in \mathbb{N}$, consiste em pôr

$$a^r = a^{\frac{m}{n}} = \sqrt[n]{a^m}.$$

Os detalhes complementares desta última definição podem ser encontrados em (LIMA et al., 1997).

Quanto ao significado da potência a^x para $x \in \mathbb{R}$, está além do escopo desse trabalho, considerando-se válidas as mesmas propriedades das potências com expoente racional vistas anteriormente.

Feitas as considerações a respeito das potências e propriedades segue a definição da Função Exponencial. Seja a um número real positivo, que suporemos sempre diferente de 1, e $\mathbb{R}^+$ o conjunto dos números reais positivos. A ***Função Exponencial*** de base a, $f : \mathbb{R} \to \mathbb{R}^+$, indicada pela notação $f(x) = a^x$, deve ser definida de modo a ter as seguintes propriedades, para quaisquer x, $y \in \mathbb{R}$:

1. $a^x \cdot a^y = a^{x+y}$;
2. $a^1 = a$;
3. $x < y \Rightarrow a^x < a^y$ quando $a > 1$ e $x < y \Rightarrow a^y < a^x$ quando $0 < a < 1$.

Não é de interesse deste trabalho aprofundar o tema Função Exponencial. A abordagem feita aqui teve o intuito de servir de base para algumas propriedades da Função Logarítmica abordadas no texto. Portanto a caracterização e as demonstrações serão omitidas e podem ser encontradas em (LIMA et al., 1997).

Logaritmos e propriedades

De acordo com (LEONARDO, 2016): “Dados os números reais positivos a e b com $a \neq 1$, o **logaritmo** de b na **base** a é o número real x tal que $a^x = b$. Ou seja:

$$\log_a b = x \iff a^x = b.$$

O número b é conhecido por **logaritmando**”. Algumas consequências da definição são:

- $\log_a 1 = 0$
- $\log_a a = 1$
- $\log_a a^n = n$
- $a^{\log_a n} = n$
- $\log_a m = \log_a n \iff m = n$

Satisfeitas as condições de existência de um logaritmo, as propriedades operatórias dos logaritmos são:

Logaritmo do produto: $\log_a (b \cdot c) = \log_a b + \log_a c$.

Demonstração 2. *Definindo-se,*

$$\log_a (bc) = x, \quad \log_a b = y \quad e \quad \log_a c = z.$$

Essas respectivas equações são equivalentes a escrever:

$$a^x = bc, \quad a^y = b \quad e \quad a^z = c.$$

Logo pode-se escrever:

$$a^x = a^y a^z = a^{y+z}$$

Segue-se que $x = y + z$ e portanto,

$$\log_a (bc) = \log_a b + \log_a c$$

Logaritmo do quociente: $\log_a \left(\frac{b}{c}\right) = \log_a b - \log_a c$.

Demonstração 3. *Analogamente a Demonstração 2, tem-se:*

$$\log_a \left(\frac{b}{c}\right) = x, \quad \log_a b = y \quad e \quad \log_a c = z.$$

Que são, respectivamente, equivalentes a,

$$a^x = \frac{b}{c}, \quad a^y = b \quad e \quad a^z = c$$

Daí,

$$a^x = a^{y-z} \iff x = y - z$$

Portanto,

$$log_a\left(\frac{b}{c}\right) = \log_a b - \log_a c$$

Logaritmo de uma potência: $\log_a b^n = n \cdot \log_a b$.

Demonstração 4. *Segue uma consequência imediata da Demonstração 1, ou seja,*

$$\log_a b^n = \log_a (b \cdot b \cdot \quad \ldots \quad \cdot b) = \underbrace{\log_a b + \log_a b + \cdots + \log_a b}_{n \quad vezes} = n \cdot \log_a b$$

Função Logarítmica

Sendo $\mathbb{R}_+^*$ o conjunto dos números reais positivos, segundo (LEONARDO, 2016), "Uma função $f : \mathbb{R}_+^* \to \mathbb{R}$ chama-se ***Função Logarítmica*** quando existe um número real a, com $a > 0$ e $a \neq 1$, tal que $f(x) = \log_a x$ para todo $x \in \mathbb{R}_+^*$".

Exemplo 2.13. Seja $f : \mathbb{R}_+^* \to \mathbb{R}$ definida por $f(x) = \log_2 x$.

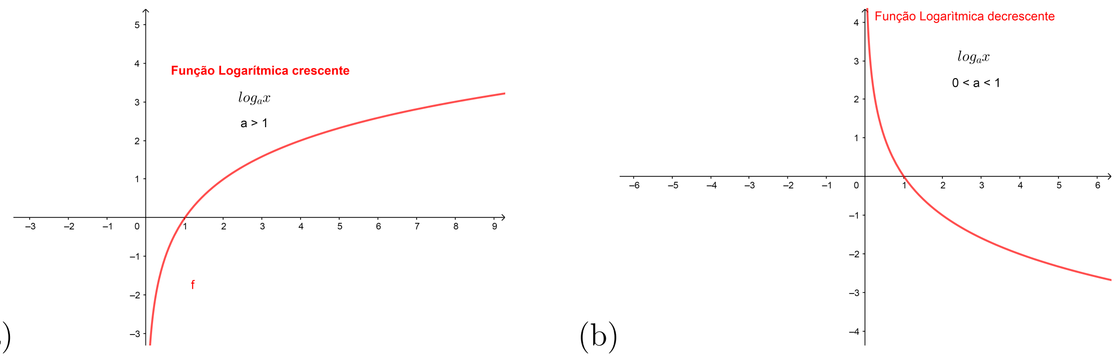
Exemplo 2.14. Seja $g : \mathbb{R}_+^* \to \mathbb{R}$ definida por $g(x) = \log_{0,3} x$.

Exemplo 2.15. Seja $h : \mathbb{R}_+^* \to \mathbb{R}$ definida por $h(x) = \log_{\frac{1}{2}} x$.

Gráfico da Função Logarítmica

O gráfico da ***Função Logarítmica*** representada por $f(x) = \log_a(x)$ definida anteriormente, possui o eixo O_y como assíntota vertical e intersecta o eixo O_x no ponto $(1, 0)$.

Figura 6 – Gráficos de Funções Logarítmicas



(a) (b)

Fonte: O próprio autor

O **crescimento** e **decrescimento** da ***Função Logarítmica*** também podem ser observados nos gráficos (a) e (b) acima.

Se $a > 1$, a função f é **crescente**.

Demonstração 5. *Seja $a > 1$, considere $x_1, x_2 \in \mathbb{R}_+^*$ e suponhamos $x_1 < x_2$. Podemos escrever*

$$f(x_1) = \log_a x_1 \iff a^{f(x_1)} = x_1 \text{ e } f(x_2) = \log_a x_2 \iff a^{f(x_2)} = x_2.$$

Daí,

$$a^{f(x_1)} < a^{f(x_2)} \iff f(x_1) < f(x_2)$$

Se $0 < a < 1$, a função f é **decrescente**.

Demonstração 6. *É análoga à Demonstração 5.*

É importante destacar que em (a) a função é estritamente crescente no intervalo $(0, +\infty)$ e em (b) estritamente decrescente no mesmo intervalo.

3 Algoritmos de Ordenação e Complexidade

O início deste capítulo aborda alguns conceitos básicos sobre algoritmos para posteriormente serem definidos e exemplificados os ***Algoritmos de Ordenação***, também realizar-se-á a análise da complexidade (desempenho) dos Algoritmos de Ordenação aqui descritos. Vale lembrar que não haverá descrição de todos os algoritmos, visto que são inúmeros e seria desnecessário para o que se propõe nesta pesquisa. Sendo assim, serão analisados apenas os Algoritmos de Ordenação: *Bubble Sort*, *Selection Sort* e *Quick Sort*. Quanto à complexidade, também chamada de desempenho ou custo, serão observados, para um vetor de n elementos, o melhor caso, quando o vetor já está ordenado, o pior caso, quando o vetor está ordenado ao contrário e o caso médio, que calcula em média o tempo para se percorrer $n/2$ elementos do vetor para obter o elemento procurado na ordenação. Tais situações serão melhor esclarecidas no decorrer do trabalho, uma vez que no *Quick Sort* os casos de desempenho se apresentam de forma diferente dos outros dois algoritmos citados. Por fim, serão analisadas as complexidades graficamente comparando-as por meio das funções que as representam utilizando a análise assintótica e também será realizada uma breve abordagem sobre fluxogramas.

3.1 Conceitos básicos

Um dos primeiros contatos que temos com algoritmos é utilizando o **Algoritmo de Euclides** para divisão entre números naturais, contudo, quando ocorrem esses primeiros contatos, não há ainda uma relação de tal algoritmo com tecnologia computacional e nem clareza quanto ao conceito mais amplo de algoritmo.

O significado de algoritmo que se apresenta nos primeiros contatos do estudante com tal conceito é mais relacionado com conceitos e procedimentos matemáticos, porém a definição é mais abrangente. Pode-se definir algoritmo como "um conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema ou de uma classe de problemas, em um número finito de etapas (RIBEIRO, 2018, p. 32)". Um algoritmo matemático pode ser passível de programação computacional e existem algoritmos não-computacionais, como por exemplo: tomar um banho. De acordo com (CORMEN, 2002), "um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou um conjunto de valores como **entrada** e produz algum valor ou conjunto de valores como **saída**". O mesmo autor defende que um algoritmo é uma ferramenta para resolver um problema computacional bem especificado e problemas de ordenação são comuns, a ordenação é uma operação fundamental na ciência da computação. Problemas de busca na internet, compras de produtos online e alocação de recursos na indústria são alguns exemplos onde se podem necessitar do uso de algoritmos. Observou-se, no Capítulo 1, que uma das competências sugeridas na BNCC é estimular o estudante desenvolver o pensamento computacional por meio da interpretação e elaboração de algoritmos.

Mesmo que se tenha, por hipótese, computadores com boa velocidade e boa memória, é necessário que haja métodos mais fáceis de serem implementados e rápidos na resolução dos problemas computacionais. Significa que um algoritmo mais eficiente torna o tempo de execução e o espaço na memória mais viáveis. Daí, para resolver o mesmo problema computacional podem ser usados métodos que diferem na eficiência e que independem do hardware utilizado. Serão avaliados posteriormente Algoritmos de Ordenação que utilizam métodos de ordenação por **comparação** sendo alguns mais simples, porém menos eficientes e outros de implementação mais complexa com melhor eficiência.

Uma ordenação por **comparação simples** realiza ou não a troca dos elementos dos vetores dois a dois após compará-los, é o caso de uma ordenação por **seleção**. Nele, para uma ordenação crescente, percorre o vetor até encontrar o menor valor e coloca-o na primeira posição, depois percorre novamente o vetor em busca do segundo menor valor e localiza-o na segunda posição e assim sucessivamente até que o vetor fique ordenado. Como exemplo dessa categoria de algoritmo temos o ***Selection Sort*** e um que utiliza ordenação por comparação mais simples do que por seleção e está aqui exemplificado pelo ***Bubble Sort***. Apesar de utilizarem um paradigma de ordenação semelhante, visto que após realizadas as comparações dos elementos se coloca os mesmos na posição e ordem almejadas, não utilizam o mesmo procedimento, e por isso é importante que conheçamos mais de um algoritmo dessa categoria. Outro algoritmo que utiliza ordenação por comparação é o ***Quick Sort***, porém com uma técnica chamada **divisão e conquista** que consiste em dividir um problema a ser resolvido em problemas menores, que podem ser divididos em partes ainda menores, encontram-se as soluções dos problemas menores e combina-se as soluções dos problemas menores em uma solução global que resolve o problema inicial. O algoritmo de ordenação mais conhecido que se utiliza desse método é o *Merge Sort*, contudo a escolha nesta pesquisa foi pelo ***Quick Sort*** e teremos seus detalhes colocados mais adiante.

Para realizar-se a comparação da eficiência dos algoritmos será utilizada a **notação assintótica** ou notação **O-grande**, representada pelo símbolo O. Tal notação **O-grande** refere-se a um limite assintótico superior. Vale salientar que ao analisar assintoticamente os algoritmos de ordenação, considera-se o termo de maior crescimento da lei de formação da função que representa o desempenho do Algoritmo de Ordenação e que não é intuito do escopo desse trabalho um aprofundamento tanto no que diz respeito à representação da notação assintótica acima citada como o motivo da escolha do termo de maior crescimento que nos referimos anteriormente.

> Quando observamos tamanhos de entrada grandes o suficiente para tornar relevante apenas a ordem de crescimento do tempo de execução, estamos estudando a **eficiência assintótica** dos algoritmos. Ou seja, estamos preocupados com a maneira como o tempo de execução de um algoritmo aumenta com o tamanho da entrada no *limite*, à medida que o tamanho da entrada aumenta indefinidamente. Em geral, um algoritmo que é assintoticamente mais eficiente será a melhor escolha para todas as entradas, exceto as muito pequenas. (CORMEN, 2002, Parte I, p. 51)

O mesmo autor acrescenta que a **eficiência assintótica** é definida em termos de funções cujo domínio é $\mathbb{N}$, contudo, às vezes é conveniente abusar da notação assintótica para um subconjunto de $\mathbb{N}$ ou ao domínio dos números reais.

Para uma dada função $g(n)$, denotamos por $O(g(n))$ o conjunto de funções $O(g(n)) = \{$*f(n): existem constantes* c e n_o *tais que* $0 \leq f(n) \leq c \cdot g(n)$ *para todo* $n \geq n_o\}$.

Exemplo 3.1. Considere as funções $f(n) = n^2 + 5n + 1$ e $g(n) = n^2$.
Tem-se, para $n \geq 1$, $n^2 + 5n + 1 \leq n^2 + 5n^2 + n^2 = 7 \cdot n^2 \Longrightarrow f(n) \leq 7 \cdot g(n)$.
Portanto,

$$f(n) \in O(g(n))$$

.

Segundo (CORMEN, 2002, p. 35), "para indicar que uma função $f(n)$ é membro de $O(g(n))$, escrevemos $f(n) = O(g(n))$".

Exemplo 3.2. Considere as funções $f(n) = 100 \log_2 n + 2n \cdot \log_2 n - 10n$, $h(n) = \log_2 n$ e $g(n) = n \cdot \log_2 n$. Mostraremos que $f(n)$ é assintoticamente representada na notação Big-O por $O(n \cdot \log_2 n)$. De fato, as funções $h(n)$ e $g(n)$ são estritamente crescentes para todo $n \geq 2$, daí pode-se escrever,

$$\log_2 n^2 \leq \log_2 n^n.$$

E assim, pode-se afirmar que $100 \log_2 n \leq 50n \cdot \log_2 n$ e somando-se $2n \log_2 n$ a esta desigualdade ambos lados tem-se,

$$100 \log_2 n + 2n \cdot \log_2 n \leq 50n \cdot \log_2 n + 2n \cdot \log_2 n.$$

Subtraindo-se $-10n$ ambos lados e sabendo que $-10n \leq n \cdot \log_2 n$ tem-se,

$$100 \log_2 n + 2n \cdot \log_2 n - 10n \leq 50n \cdot \log_2 n + 2n \cdot \log_2 n + n \cdot \log_2 n.$$

Segue-se que,

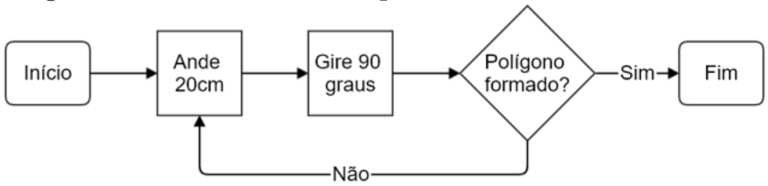
$$100 \log_2 n + 2n \cdot \log_2 n - 10n \leq 53 \cdot n \cdot \log_2 n.$$

Logo, $0 \leq f(n) \leq c \cdot g(n) \quad$ e $\quad f(n) = O(g(n)) = O(n \cdot \log_2 n)$.

Outro conceito também utilizado no estudo de algoritmos é o de ***fluxograma*** que é um diagrama escrito em uma notação gráfica simples, usado para representação visual de um algoritmo. Como exemplo simples de um fluxograma, podemos observar a figura a seguir que trata da construção de um quadrado.

Figura 7 – Fluxograma indicando um algoritmo para formar um quadrado.

Tente desenhar no seu caderno, a figura que o fluxograma abaixo propõe.
Os giros deverão ser feitos apenas no sentido anti-horário.



Fonte:<https://novaescola.org.br/plano-de-aula/5129/interpretando-fluxogramas>

Existem algumas normas para a construção de um fluxograma que serão abordadas em momento oportuno e também serão observados mais exemplos sobre o tema no final do capítulo.

3.2 Algoritmos de ordenação

Serão analisados a seguir dois Algoritmos de Ordenação que fazem uso de estratégias de comparação distintas, são eles: *Bubble Sort* e *Selection Sort*. Apesar de utilizarem método afim, ou seja, paradigmas semelhantes, os mesmos diferem no procedimento e têm sua eficiência diferindo no melhor caso, contudo se faz necessário analisá-los no pior caso e no caso médio observando as diferenças nos desempenhos.

> Muitas vezes o caso médio é quase tão ruim quanto o pior caso [. . .] Em média, metade do elementos em A$[1, 2, 3, \ldots, j-1]$ são menores que A$[j]$, e metade são maiores. [. . .] Se desenvolvermos o tempo de execução do caso médio resultante, ele será uma função quadrática do tamanho da entrada, exatamente como o tempo de execução do pior caso. (CORMEN, 2002, Parte I, p. 20)

Apesar de se fazer referência no texto em alguns exemplos, o caso médio não será prioritário em nossas análises, pois a análise assintótica tem como principal preocupação um número muito grande de elementos de um vetor a ser ordenado avaliando o pior caso, além disso o caso médio é considerado difícil de obter sua complexidade visto que nem sempre é claro o que seria o caso médio em relação à distribuição dos elementos do vetor a ser ordenado, como ocorre com o *Quick Sort*.

3.2.1 *Bubble Sort*

O Algoritmo de Ordenação ***Bubble Sort*** é considerado um dos mais simples, também é conhecido por Ordenação por Flutuação, daí o nome ***Bubble*** cuja tradução para o Português é **Bolha**. Seu procedimento se dá da seguinte maneira: para uma ordenação crescente, iniciando-se com o primeiro, cada elemento do vetor se compara com o posterior, caso este seja maior, muda de posição com o mesmo, caso não o seja, a troca de posição não é feita e passa-se para o próximo par de comparação. Para uma ordenação decrescente, o elemento se compara com o posterior, caso seja maior, não muda sua posição atual. Segue-se para o próximo par de comparação e assim sucessivamente até que o vetor esteja ordenado. Além da simplicidade outra vantagem desse Algoritmo de Ordenação é a estabilidade, mas a lentidão o deixa em desvantagem em relação a outros Algoritmos de Ordenação. É indicando para pequenos vetores e contraindicado para vetores com número grande de elementos.

Exemplo 3.3. Considere o vetor $A = (5, 4, 3, 2, 1)$, que possui o número de elementos $n = 5$. O objetivo é colocá-lo em ordem crescente com a estratégia de comparação do *Bubble Sort*. O vetor A caracteriza o pior caso do *Bubble Sort*.
Os procedimentos se dão da seguinte maneira:
Passo I. 5 compara com 4 e troca de posição, pois 5>4;
Passo II. 5 compara com 3 e troca de posição, pois 5>3;
Passo III. 5 compara com 2 e troca de posição, pois 5>2;
Passo IV. 5 compara com 1 e troca de posição, pois 5>1.
Agora ocorre o mesmo com o 4.
Passo V. 4 compara com 3 e troca de posição, pois 4>3;
Passo VI. 4 compara com 2 e troca de posição, pois 4>2;
Passo VII. 4 compara com 1 e troca de posição, pois 4>1.
Passo VIII. 3 compara com 2, como $3 < 2$, troca de posição com este.
Passo IX. 3 compara com 1 e também troca de posição com ele.
Passo X. 2 compara com 1 e troca de posição.

Outra forma de representar o procedimento está descrito na tabela a seguir.

Tabela 2 – Exemplo do *Bubble Sort* com 5 elementos, $n = 5$, representando o pior caso.

VETOR	Nº de comparações
5-4-3-2-1	4 comparações (5 e 4, 5 e 3, 5 e 2, 5 e 1).
4-3-2-1-5	3 comparações (4 e 3, 4 e 2, 4 e 1).
3-2-1-4-5	2 comparações (3 e 2, 3 e 1).
2-1-3-4-5	1 comparação (2 e 1).
1-2-3-4-5	$(n-1)$ iterações: vetor ordenado.

Fonte: Adaptado de <https://www.ic.unicamp.br/~oliveira/doc/mc102_2s2004/aula_Ordenacao1.pdf>

Seja, por exemplo, C_n a quantidade de comparações quando se retira o elemento a de um vetor de n elementos. Portanto tem-se $C_n = n - 1$ comparações. Com esta ideia pode-se escrever

$$\sum_{i=1}^{n} C_i = (n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \tag{3.1}$$

comparações no pior caso. Sua eficiência(complexidade) é, assintoticamente, representada por $O(n^2)$. No melhor caso do *Bubble Sort*, quando o vetor já se encontra ordenado, teremos apenas $(n-1)$ iterações e sua eficiência(complexidade) é representada por $O(n)$. No pior caso temos uma complexidade representada por um função quadrática e no melhor caso por uma função linear. Vejamos outro exemplo com um vetor desordenado e com número maior de elementos.

Exemplo 3.4. Dado o vetor $w = (7, 12, 5, 0, 21, 3, 6, 1, 2, 9)$, com o número de elementos $n = 10$, utilizando a estratégia do Algoritmo *Bubble Sort*, teremos a seguinte configuração de comparações para uma ordenação crescente:
Passo I. 7 compara com 12 e não troca de posição, já que 7< 12;
Passo II. 12 compara com com 5 e troca de posição, visto que 12>5;
Passo III. 12 compara com 0 e troca de posição, uma vez que 12>0;
Passo IV. 12 compara com 21 e não troca de posição, pois 12<21;
Passo V. 21 compara com 3 e muda de posição, já que 21>3;
Passos VI a IX. 21 compara com 6, 1, 2 e 9 e muda de posições com os mesmos, visto que 21>6, 21>1, 21>2 e 21>9.
Continuando com nossas comparações temos:
Passos X e XI 7 compara com 5, 0 e troca de posição com os mesmos, pois 7>5 e 7>0;
Passo XII. 7 compara com 12, que ficou na posição do 21, e não troca de posição pois 7<12;
Passos XIII a XVI. 12 compara com 3, 6, 1, 2, 9 e troca de posição com os mesmos, haja vista que 12>3, 12>6, 12>1 e 12>2;
Temos 12 e 21 nas posições corretas.
Passo XVII. 5 compara com 0 e troca de posição, já que 5>0;
Passo XVIII. 5 compara com 7 e não troca de posição, porque 5<7;
Passos XIX a XXII. 7 compara com 3, 6, 1, 2 e troca de posição com estes;
Passo XXIII. 7 compara com 9 e não troca de posição, pois 7<9.
9 já foi comparado com 12 e 21, e fica na antepenúltima posição.
Continuaremos com as comparações até o **Passo XLV** de modo que tenhamos o vetor ordenado.

Apresenta-se na tabela abaixo, como no exemplo anterior, uma representação dos procedimentos aqui realizados com o vetor w, cuja configuração inicial não está em ordem crescente ou decrescente.

Tabela 3 – Exemplo do *Bubble Sort* com vetor de 10 elementos, $n = 10$, representando o caso médio.

VETOR	Nº de comparações
7-12-5-0-21-3-6-1-2-9	9 comparações (7 e 12, 12 e 5, 12 e 0, 12 e 21,21 e 3, 21 e 6, 21 e 1, 21 e 2, 21 e 9).
7-5-0-12-3-6-1-2-9-21	8 comparações (7 e 5, 7 e 0, 7 e 12, 12 e 3, 12 e 6, 12 e 1, 12 e 2, 12 e 9).
5-0-7-3-6-1-2-9-12-21	7 comparações (5 e 0, 5 e 7, 7 e 3, 7 e 6, 7 e 1, 7 e 2, 7 e 9).
0-5-3-6-1-2-7-9-12-21	6 comparações (0 e 5, 5 e 3, 5 e 6, 6 e 1, 6 e 2, 6 e 7).
0-3-5-1-2-6-7-9-12-21	5 comparações (0 e 3, 3 e 5, 5 e 1, 5 e 2, 5 e 6).
0-3-1-2-5-6-7-9-12-21	4 comparações (0 e 3, 3 e 1, 3 e 2, 3 e 5).
0-1-2-3-5-6-7-9-12-21	3 comparações (0 e 1, 1 e 2, 2 e 3).
0-1-2-3-5-6-7-9-12-21	2 comparações (0 e 1, 1 e 2).
0-1- 2-3-5-6-7-9-12-21	1 comparação (0 e 1).
0-1-2-3-5-6-7-9-12-21	$(n-1)$ iterações: vetor ordenado.

Fonte: Adaptado de <https://www.ic.unicamp.br/~oliveira/doc/mc102_2s2004/aula_Ordenacao1.pdf>

Considerando um vetor de n elementos, analogamente a 3.1, temos o total de comparações C_n, assim como no pior caso, representado por

$$\sum_{i=1}^{n} C_i = (n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \tag{3.2}$$

comparações para o caso médio, onde o vetor não se encontra totalmente em ordem crescente ou decrescente. Sua complexidade é, assintoticamente, representada por $O(n^2)$.

3.2.2 *Selection Sort*

O termo ***Selection Sort*** vem do inglês e significa **Ordenação por seleção**, também utiliza o método ou paradigma da comparação por seleção ou seleção direta. Nela, a ideia é ordenar a lista selecionando em cada iteração os menores itens e ordenando-os da esquerda para direita em caso de ordem crescente. Quando o menor elemento da lista é localizado na 1ª posição a iteração ocorre para localizar o segundo menor elemento da lista na 2ª posição e assim sucessivamente até que a lista se ordene em ordem crescente. Algumas vantagens desse algoritmo são: fácil implementação e uso de pouca memória. Entre as desvantagens temos como principal seu desempenho, é ruim em todos os casos: melhor, médio e pior, sendo indicado apenas para pequenas listas e não é um algoritmo estável.

Exemplo 3.5. Consideremos a lista(vetor) $B = (5, 3, 1, 2, 4)$, com $n = 5$ elementos, a qual será ordenada de acordo com os seguintes procedimentos do *Selection Sort*.

Passo I. 5 compara com 3, 3 é menor e troca de posição com 5, assim 3 vai para 1ª posição;

Passo II. 3 compara com 1, 1 é menor e troca de posição com 3, ou seja, 1 fica na 1ª posição e 3 na 3ª posição;

Passo III. 1 compara com 2, 1 é menor e continua sem trocar;
Passo IV. 1 compara com 4; 1 é menor e continua sem trocar.
O 1 é o menor elemento e ocupa a 1ª posição, após isso temos:
Passo V. 5 compara com 3, 3 é menor e muda de posição com 5;
Passo VI. 3 compara com 2, 2 é menor e muda de posição com 3;
Passo VII. 2 compara com 4, 2 é menor e não muda de posição.
O 2 irá para a 2º posição no lugar do 3, e o 3 irá para 4ª posição.
Em seguida teremos
Passo VIII. 5 compara com 3, 3 é menor e muda de posição com 5;
Passo IX. 3 compara com 4, 3 é menor e não muda de posição.
O 3 irá para a 3ª posição no lugar do 5, que está na posição inicial do 1 e o 5 vai para a 4ª posição, onde estava o 3. Realizada a troca, teremos a última comparação:
Passo X. 5 compara com 4, 4 é menor e muda de posição com 5. O 4 vai para 4ª posição e o 5 para 5ª posição.

A representação dos procedimentos realizados resumem-se na tabela abaixo.

Tabela 4 – Exemplo do *Selection Sort* com 5 elementos, $n = 5$, representando o caso médio.

VETOR	**Nº de comparações**
5-3-1-2-4	4 comparações (5 e 3, 3 e 1, 1 e 2, 1 e 4)
1-5-3-2-4	3 comparações (5 e 3, 3 e 2, 2 e 4)
1-2-5-3-4	2 comparações (5 e 3, 3 e 4)
1-2-3-5-4	1 comparação (5 e 4)
1-2-3-4-5	$(n-1)$ iterações: vetor ordenado

Fonte: Adaptado de <https://www.ic.unicamp.br/~oliveira/doc/mc102_2s2004/aula_Ordenacao1.pdf>

Utilizando o mesmo raciocínio de 3.1 e 3.2, para um vetor de n elementos temos o total de Comparações C_n dado por

$$\sum_{i=1}^{n} C_i = (n-1) + (n-2) + ... + 1 = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \tag{3.3}$$

comparações **em qualquer caso**. Sua eficiência é representada assintoticamente por $O(n^2)$.

Exemplo 3.6. Dado o vetor $C = (5, 9, 16, 57, 7, 2, 3)$, com $n = 7$ elementos, a ordenação com a estratégia do *Selection Sort* se dá da seguinte maneira:
Passo I. 5 compara com 9, como 5 é menor não muda de posição com o 9;
Passo II a IV. 5 compara com 16, 57 e 7, não mudando de posição com nenhum deles;
Passo V. 5 compara com 2, como 2 é menor muda de posição com 5;
Passo VI. 2 compara com 3, como 2 é menor não muda de posição.
Teremos, então, o 2 na 1ª posição.
Em seguida, continuamos as comparações a partir do 9 que está na 2ª posição.

Passo VII e VIII. 9 compara com 16 e 57, mas não muda de posição, pois é menor que estes;
Passo IX. 9 compara com 7, como 7 é menor, muda de posição com o 9;
Passo X. 7 compara com 5, como 5 é menor, muda de posição com o 7;
Passo XI. 5 compara com 3 e muda de posição com o mesmo, pois 5>3.
Assim, temos o 3 na 2ª posição. Fazemos agora as comparações começando a iteração com o 16.
Passo XII. 16 compara com 57 e não muda de posição;
Passo XIII. 16 compara com 9 e troca de posição com o mesmo, porque 16>9;
Passo XIV. 9 compara com 7 e troca de lugar com este;
Passo XV. 7 compara com 5 e troca de lugar com ele, pois 5<7.
Temos agora a seguinte ordem: 2, 3, 5, 57, 16, 9 e 7. A partir daí,
Passo XVI. 57 compara com 16 e troca de posição;
Passo XVII. 16 compara com com 9 e também troca de posição com este;
Passo XVIII. 9 compara com 7 e troca de posição com o mesmo.
Ficamos com o 7 na 4ª posição e o vetor agora ficou ordenado assim: 2, 3, 5, 7, 57, 16 e 9.
Passo XIX. O 57 compara com 16 e troca de posição;
Passo XX. 16 compara com 9 e troca de posição.
O 9 fica na 5ª posição e o vetor fica na seguinte ordem: 2, 3, 5, 7, 9, 57, 16.
Passo XXI. 57 compara com 16 e troca de posição, finalizando a ordenação do vetor.

Todo o procedimento está resumido na tabela a seguir.

Tabela 5 – Exemplo do *Selection Sort* com $n = 7$ elementos, representando o caso médio.

VETOR	**Nº de comparações**
5-9-16-57-7-2-3	6 comparações (5 e 9, 5 e 16, 5 e 57, 5 e 7, 5 e 2, 2 e 3)
2-9-16-57-7-5-3	5 comparações (9 e 16, 9 e 57, 9 e 7, 7 e 5, 5 e 3)
2-3-16-57-9-7-5	4 comparações (16 e 57, 16 e 9, 9 e 7, 7 e 5)
2-3-5-57-16-9-7	3 comparação (57 e 16, 16 e 9, 9 e 7)
2-3-5-7-57-16-9	2 comparações (57 e 16, 16 e 9)
2-3-5-7-9-57-16	1 comparação (57 e 16)
2-3-5-7-9-16-57	$(n-1)$ iterações: vetor ordenado

Fonte: Adaptado de <https://www.ic.unicamp.br/~oliveira/doc/mc102_2s2004/aula_Ordenacao1.pdf>

Para um vetor de n elementos, de modo análogo a 3.3 temos

$$\sum_{i=1}^{n} C_i = (n-1) + (n-2) + ... + 1 = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \tag{3.4}$$

comparações **em qualquer caso**, com complexidade representada por $O(n^2)$.

3.2.3 *Quick Sort*

Será visto nesta seção um Algoritmo de Ordenação chamado ***Quick Sort***. O mesmo utiliza uma estratégia de ordenação por comparação bastante diferente das duas abordadas anteriormente, conhecida por divisão e conquista. Neste tipo de Algoritmo de Ordenação o vetor é dividido em duas partes a partir de um elemento denominado pivô, antes do pivô são colocados os elementos menores ou iguais ao mesmo e depois dele os elementos maiores. O processo se repete em cada uma das partes sucessivamente até que se obtenha partes com apenas um elemento. Para finalizar, acontece o que chama-se de conquista, os resultados são combinados e obtém-se o resultado esperado. É um algoritmo bastante popular, considerado rápido e eficiente, utiliza técnicas de recursão sendo considerado complexo e não estável. A escolha do pivô e forma de particionamento são situações que não serão levadas em conta neste estudo, interessando apenas a análise e comparação com os demais algoritmos em estudo. No esquema a seguir apresenta-se um exemplo de como seria a ordenação utilizando a estratégia de divisão e conquista do ***Quick Sort***.

Exemplo 3.7. Dado o vetor $Q = (5, 3, 4, 9, 1, 7, 6, 8, 2)$ nesta ordem, sendo $n = 9$. Uma das possíveis ordenações utilizando o ***Quick Sort*** seria:

$$\boxed{5}\ \boxed{3}\ \boxed{4}\ \boxed{9}\ \boxed{1}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{2}$$

Considere o pivô escolhido 5.

Considere r a última posição dos elementos que formam o vetor e $p \neq r$ qualquer outra posição que um elemento ocupa no vetor. Após a escolha do pivô, este fica na posição r e para que os valores menores que o pivô fiquem antes dele e os valores maiores que ele fiquem depois, as comparações ocorrem da seguinte maneira: são criadas duas variáveis i e j representando as posições dos valores, estas variáveis serão usadas para percorrer o vetor de entrada de modo que cada posição i será percorrida da esquerda para direita, para localizar elementos com valores menores que o pivô, e cada posição j também será percorrida da esquerda para a direita, até localizar elementos maiores que o pivô. Inicia-se com $i = j - 1$, ao tempo em que j vai percorrendo o vetor, na busca dos valores maiores que o pivô, i vai logo após localizando os valores menores que o pivô. À medida que forem localizando, respectivamente, valores maiores e menores, passam para posição imediatamente posterior até que j localize um valor menor que o pivô e i localize um valor maior que o pivô, neste caso o valor de posição j troca de lugar com o valor de posição i. O procedimento continua até que todos os valores menores que o pivô estejam numa posição $p \leq i$ e os valores maiores que o pivô estejam numa posição $i < p \leq j$. Feito isso, o pivô ocupará a posição $i + 1$. Com relação ao vetor Q teremos:

$$\underbrace{\phantom{\boxed{0}}}_{i\rightarrow}\ \underbrace{\boxed{3}}_{j\rightarrow}\ \boxed{4}\ \boxed{9}\ \boxed{1}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\underbrace{\boxed{3}}_{i\rightarrow}\ \underbrace{\boxed{4}}_{j\rightarrow}\ \boxed{9}\ \boxed{1}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\boxed{3}\ \underbrace{\boxed{4}}_{i\rightarrow}\ \underbrace{\boxed{9}}_{j\rightarrow}\ \boxed{1}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\boxed{3}\ \boxed{4}\ \underbrace{\boxed{9}}_{i\rightarrow}\ \underbrace{\boxed{1}}_{j\rightarrow}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

Agora o valor 1 muda de posição com o valor 9 e continuamos com o vetor no seguinte formato.

$$\boxed{3}\ \boxed{4}\ \underbrace{\boxed{1}}_{i\rightarrow}\ \underbrace{\boxed{9}}_{j\rightarrow}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\boxed{3}\ \boxed{4}\ \boxed{1}\ \underbrace{\boxed{9}}_{i\rightarrow}\ \underbrace{\boxed{7}}_{j\rightarrow}\ \boxed{6}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\boxed{3}\ \boxed{4}\ \boxed{1}\ \underbrace{\boxed{9}}_{i\rightarrow}\ \boxed{7}\ \underbrace{\boxed{6}}_{j\rightarrow}\ \boxed{8}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\boxed{3}\ \boxed{4}\ \boxed{1}\ \underbrace{\boxed{9}}_{i\rightarrow}\ \boxed{7}\ \boxed{6}\ \underbrace{\boxed{8}}_{j\rightarrow}\ \boxed{2}\ \underbrace{\boxed{5}}_{r}$$

$$\boxed{3}\ \boxed{4}\ \boxed{1}\ \underbrace{\boxed{9}}_{i\rightarrow}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \underbrace{\boxed{2}}_{j\rightarrow}\ \underbrace{\boxed{5}}_{r}$$

Observe que ocorreu, simultaneamente, de j localizar um valor menor que o pivô 5 e i localizar um valor maior que o pivô 5. Daí, o valor 2 muda de posição com o valor 9, ficando o vetor com nova configuração.

$$\boxed{3}\ \boxed{4}\ \boxed{1}\ \underbrace{\boxed{2}}_{i\rightarrow}\ \boxed{7}\ \boxed{6}\ \boxed{8}\ \underbrace{\boxed{9}}_{j\rightarrow}\ \underbrace{\boxed{5}}_{r}$$

Temos que j está na posição $r-1$ e já encerrou o percurso pelo vetor, de modo que o valor na posição i também não sofrerá mais nenhuma troca com algum valor de posição j. Assim, o pivô 5 de posição r ocupará a posição $i+1$ e ficaremos com dois subvetores: o primeiro antes do pivô, com elementos menores que 5 e o segundo, depois do pivô, com elementos maiores que 5.

$$\begin{array}{ccc} & \boxed{5} & \\ \boxed{3}\ \boxed{4}\ \boxed{1}\ \boxed{2} & & \boxed{7}\ \boxed{6}\ \boxed{8}\ \boxed{9} \end{array}$$

Escolhemos agora, nos subvetores, os pivôs 3 e 9, respectivamente, ocorrendo o mesmo procedimento nos subvetores quando o pivô foi 5. A configuração nos dois subvetores ficam como segue.

$$\boxed{1}\,\boxed{2}\,\boxed{3}\,\boxed{4} \qquad \boxed{7}\,\boxed{6}\,\boxed{8}\,\boxed{9}$$

Agora escolhemos nos subvetores $\boxed{1}\,\boxed{2}$ e $\boxed{7}\,\boxed{6}\,\boxed{8}$ os pivôs $\boxed{1}$ e $\boxed{6}$, respectivamente. Novamente a estratégia de divisão é efetuada e os subvetores ficam com a seguinte ordem:

$$\boxed{1}\,\boxed{2} \qquad \text{e} \qquad \boxed{6}\,\boxed{7}\,\boxed{8}$$

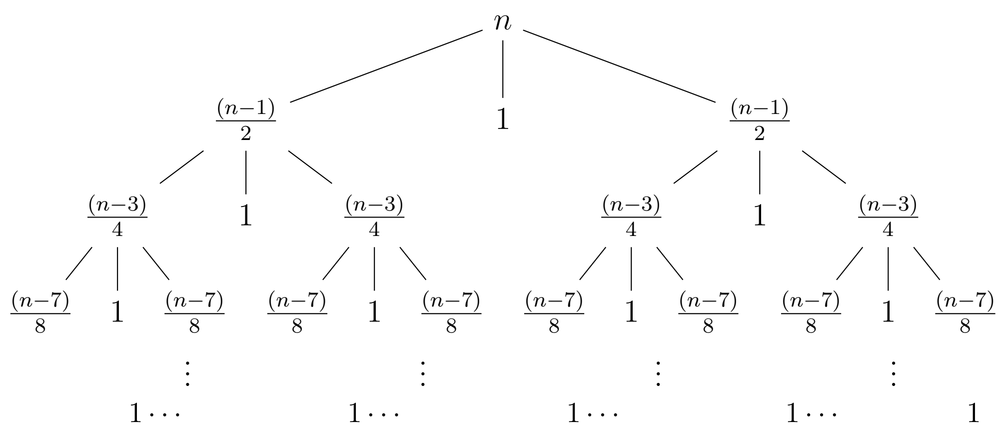
Apesar de o vetor já estar ordenado, os subvetores sofrem subdivisões até que se obtenha o último subvetor com tamanho unitário. Desse modo, no subvetor, não unitário, $\boxed{7}\,\boxed{8}$, escolhemos o pivô $\boxed{7}$. Executado novamente o procedimento de divisão do *Quick Sort*, ficando o vetor ordenado como segue.

$$\boxed{1}\,\boxed{2}\,\boxed{3}\,\boxed{4}\,\boxed{5}\,\boxed{6}\,\boxed{7}\,\boxed{8}\,\boxed{9}$$

Observe que a escolha do pivô é aleatória, mas o pivô ideal seria aquele que dividisse o vetor ou subvetor em tamanhos aproximadamente iguais, ou seja, um pivô que represente um valor mediano do vetor ou subvetor. Contudo, isso não ocorre na prática, e a escolha aleatória dos pivôs acarretam nas situações de melhor caso, pior caso ou caso médio. No melhor caso o particionamento produz segmentos(subvetores) de tamanhos iguais, já o pior caso ocorre quando o pivô é o maior (ou menor) elemento do vetor ou subvetor. O caso médio ocorre quando o particionamento é desbalanceado, como por exemplo numa partição onde todos os subproblemas estejam na proporção 1 para 8 em cada nível.

A seguir, far-se-á uma abordagem matemática, na análise da complexidade para o melhor caso do *Quick Sort*, que se mostre clara e acessível para o Ensino Médio e que auxilie os professores na obtenção da complexidade de tal algoritmo. A comparação por árvore de distribuição binária foi a escolha que jugou-se mais viável para se conjecturar a função que representa o desempenho do ***Quick Sort***.

Figura 8 – Árvore de distribuição binária do *Quick Sort*, no melhor caso, para um vetor de n elementos.



Fonte: Adaptado de <https://slideplayer.com.br/slide/364011/>

Na tabela a seguir apresenta-se uma compilação do que está representado na árvore anterior, com o intuito de facilitar a obtenção da função que representa o desempenho desse Algoritmo de Ordenação.

Tabela 6 – Resumo da distribuição binária do *Quick Sort* representada na Figura (8)

Níveis(altura)	**Nº de subproblemas**	Tamanho do subproblema
0	1	n
1	2	$(n-1)/2$
2	4	$(n-3)/4$
3	8	$(n-7)/8$
$\vdots$	$\vdots$	$\vdots$
j	2^j	$\frac{[n-(2^j-1)]}{2^j}$

Fonte: Adaptado de <https://slideplayer.com.br/slide/364011/>

Para obter a complexidade inicialmente calcula-se o trabalho $\tau(j)$ realizado no nível j utilizando a relação

$$\tau(j) = (\text{n}^{\text{o}} \text{ de subproblemas no nível})\cdot(\text{tamanho do subproblema no nível}).$$

Da igualdade anterior temos,

$$\tau(j) = 2^j \cdot \frac{[n-(2^j-1)]}{2^j} = n - 2^j + 1. \tag{3.5}$$

O número de níveis j da árvore pode ser obtida igualando $\frac{[n-(2^j-1)]}{2^j}$ a 1, que é o tamanho do último subproblema do particionamento. Daí, obtemos:

$$\frac{[n-(2^j-1)]}{2^j} = 1 \Longrightarrow 2 \cdot 2^j = n+1 \Longrightarrow j = \log_2\left(\frac{n+1}{2}\right). \tag{3.6}$$

Como o valor de j pode não ser inteiro a depender do valor de n, será mais adequado obter o número de comparações utilizando a ***função maior inteiro*** $\lfloor j \rfloor$, com $\lfloor j \rfloor \leq j$ para todo $j \in \mathbb{R}$.

Sabendo-se do valor de $\lfloor j \rfloor$ é possível calcular o trabalho total, indicado por $T(j)$, bastando somar o trabalho por nível, ao longo dos níveis, isto é:

$$T(j) = \sum_{i=1}^{\lfloor j \rfloor}(n-2^i+1) = \sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor}(n-2^i+1) = \underbrace{\sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} n}_{I} - \underbrace{\sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} 2^i}_{II} + \underbrace{\sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} 1}_{III}. \tag{3.7}$$

Do último resultado, desenvolvendo separadamente cada parcela, obtemos:

$$(I) \quad \sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} n = n \cdot \sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} 1 = n \cdot \left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor = n \cdot \lfloor \log_2 (n+1) - 1 \rfloor$$
$$\leq n \cdot \lfloor \log_2 (2n) - 1 \rfloor \leq n \cdot \lfloor \log_2 n \rfloor \leq n \cdot \log_2 n.$$

$$(II) \quad \sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} 2^i = \frac{2^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor + 1} - 1}{2-1} - 2^0 = 2^{\lfloor \log_2 (n+1) \rfloor} - 2 \leq 2^{\log_2 (n+1)} - 2 \leq n - 1 \leq n.$$

$$(III) \quad \sum_{i=1}^{\left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor} 1 = \left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor = \lfloor \log_2 (n+1) - 1 \rfloor \leq \lfloor \log_2 2n \rfloor - 1 \leq \log_2 n.$$

Em cada parcela anterior a que possui maior ordem de crescimento é $n \cdot \left\lfloor \log_2 \frac{(n+1)}{2} \right\rfloor$, pois $\log_2 n \leq n \leq n \cdot \log_2 n$. Observa-se ainda que, dada uma função $f(n) = n \cdot (\log_2 \frac{(n+1)}{2})$ e outra função $g(n) = n \cdot \log_2 n$, temos $n \cdot (\log_2 \frac{(n+1)}{2}) \leq n \cdot (\log_2 n)$. Ou seja, $0 \leq f(n) \leq c \cdot g(n)$, sendo $c = 1$. Daí, a complexidade do trabalho total $T(j)$ realizado pelo Algoritmo *Quick Sort*, para o melhor caso, é representada assintoticamente por $O(n \cdot \log_2 n)$ ou $O(n \cdot \log n)$.

Exemplo 3.8. Considere o vetor $C = (2, 5, 1, 9, 7)$, com $n = 5$, que deve ser ordenado utilizando a estratégia do *Quick Sort*.

$\boxed{2}\ \boxed{5}\ \boxed{1}\ \boxed{9}\ \boxed{7}$

Pivô escolhido 9.

$\boxed{2}\ \boxed{5}\ \boxed{1}\ \boxed{7}\ \boxed{9}$

Temos dois subvetores com elementos menores que 9, antes dele, e os maiores que ele depois, no entanto o vetor após 9 ficou vazio, pois todos os demais valores são menores que 9.

Escolhemos agora no subvetor o pivô 7.

$\boxed{2}\ \boxed{5}\ \boxed{1}\ \boxed{7}\ \boxed{9}$

Em seguida escolhemos o pivô 5 no subvetor.

$\boxed{2}\ \boxed{1}\ \boxed{5}\ \boxed{7}\ \boxed{9}$

Escolhemos agora o pivô 2.

$\boxed{1}\ \boxed{2}\ \boxed{5}\ \boxed{7}\ \boxed{9}$

O vetor ficou ordenado.

Observa-se que a escolha de um pivô que representa o maior ou menor elemento da lista ou também da sublista é algo raro nesse tipo de problema, tem-se, para tal situação, um desempenho $\mathrm{O}(n^2)$, pois a altura da distribuição será $(n-1)$ e o comportamento de ordenação é semelhante ao *Selection Sort*. Esse tipo de situação simula o pior caso do *Quick Sort*. O caso médio se aproxima do melhor caso e sua complexidade também é $O(n \cdot \log n)$ e apesar do seu desempenho no pior caso ser $O(n^2)$, na média sua performance é excelente o que o faz ser tão popular.

Comparação gráfica das complexidades dos Algoritmos de Ordenação

Observou-se no Capítulo 2 algumas funções elementares cujos gráficos utilizaremos para comparar a complexidade dos Algoritmos de Ordenação explorados neste trabalho.

Propõe-se avaliar seus desempenhos (complexidades) inicialmente numa tabela e depois graficamente onde sua visualização possa expressar melhor o comportamento assintótico das funções que os representam

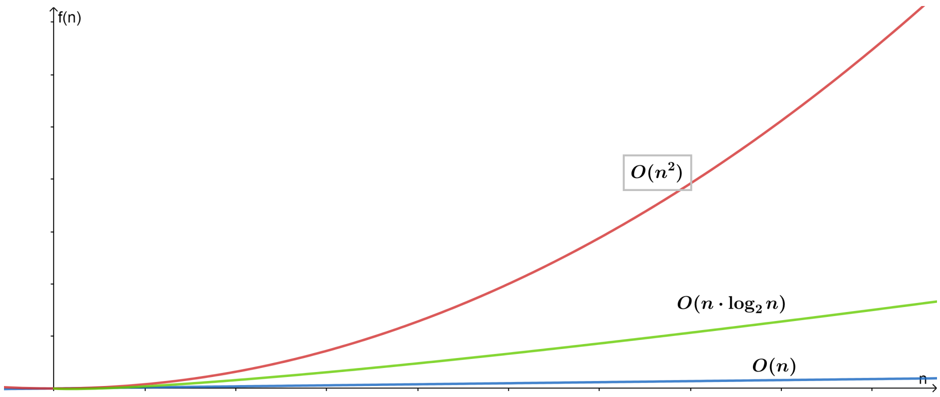
Tabela 7 – Tabela de comparação do desempenho assintótico dos Algoritmos de Ordenação

Nº de elementos	***Bubble Sort*: $O(n)$**	***Selection Sort*: $O(n^2)$**	***Quick Sort*: $O(n \cdot \log_2 n)$**
2	2	4	2
4	4	16	8
16	16	256	64
256	256	65.536	2048
1.024	1.024	1.048.576	10.240

Fonte: Produzido pelo autor

É notório na tabela anterior que o *Bubble Sort*, no melhor caso, é sempre mais eficiente que os outros dois Algoritmos de Ordenação sendo que o *Quick Sort* se mostra mais eficiente que um algoritmo de complexidade Quadrática.

Figura 9 – Gráfico de comparação assintótica dos Algoritmos de Ordenação



Fonte: Produzido pelo autor

A complexidade $O(n \cdot \log n)$, chamada de linearítmica, possui uma taxa de crescimento maior que uma complexidade linear $O(n)$ e menor que qualquer complexidade polinomial como é o caso de $O(n^2)$. Sabe-se que a análise assintótica *Big-O* compara as funções no limite superior, ou seja, quando a quantidade n de elementos do vetor é suficientemente grande. Um algoritmo é dito mais eficiente que outro quando sua complexidade possui assintoticamente uma taxa de crescimento menor, portanto conclui-se que $O(n) < O(n \cdot \log n) < O(n^2)$.

3.3 Fluxogramas

Foi iniciada uma breve abordagem sobre o tema fluxogramas o início deste capítulo e será dada uma atenção maior nesta parte do trabalho, visto que o estudo de algoritmos está bastante ligado com esse tipo de representação gráfica. Sua construção requer que siga-se algumas regras gerais como segue.

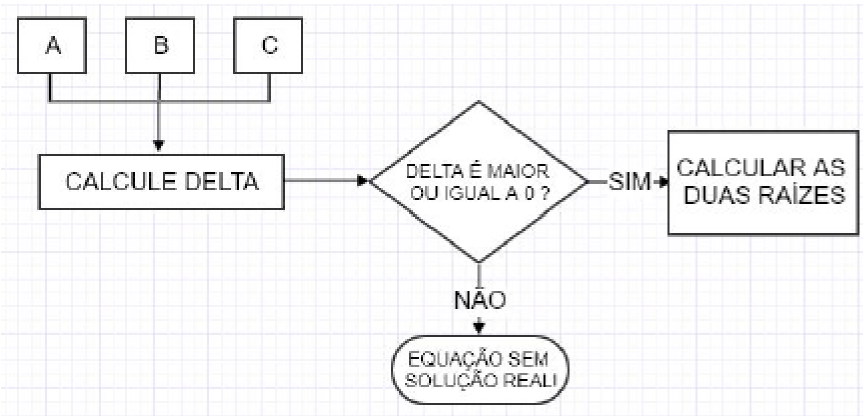
> O início e o fim do algoritmo são marcados com uma figura elíptica; as ações a serem executadas estão em retângulos; sendo que as estruturas de controle condicionais estão em losangos e indicam duas possibilidades de prosseguimento do algoritmo, uma para o caso da expressão avaliada (condição) ser verdadeira e outra para o caso de ser falsa. (FERRARI; CECHINEL, 2008)

Nesta seção, serão tratadas representações simples por fluxogramas. Quanto à todas as regras de construção, podem ser facilmente obtidas, como sugestão, em: <https://memoria.ifrn.edu.br/bitstream/handle/1044/1514/Manual%20de%20Fluxogramas%20-%20ebook%20%281%29.pdf?sequence=5&isAllowed=y>

Dentre as vantagens desse tipo de representação destacam-se: padronização na representação, descreve com maior rapidez um conjunto de tarefas e facilita a leitura e o entendimento de uma atividade.

Exemplo 3.9. Fluxograma indicando a resolução de uma equação do segundo grau.

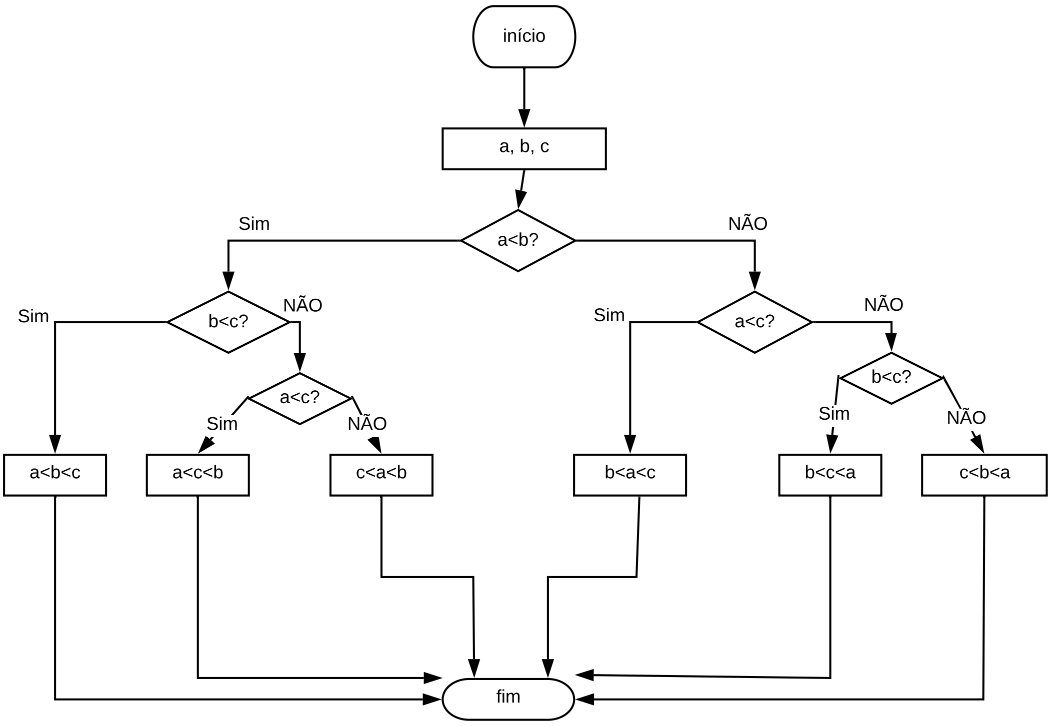
Figura 10 – Fluxograma de um algoritmo para resolver uma equação do tipo $Ax^2 + Bx + C = 0$



Fonte:<https://nova-escola-producao.s3.amazonaws.com/PNYck8d5xkQ7Ejy7S92uz64UAG54EwbxKHF5kwwWJR6cwgTfyZZjKfM2cXsq/fluxogramas-algoritmos-matematica-e-tecnologia.pdf>

Exemplo 3.10. Fluxograma indicando o algoritmo *Bubble Sort* para colocar em ordem crescente os valores da lista (a, b, c), considerando $a \neq b \neq c$.

Figura 11 – Fluxograma para o *Bubble Sort*



Fonte: Produzido pelo autor

Observe que na Figura 10 não há uma forma indicando início e fim, mas isso não interfere na aprendizagem do tema. Também vale destacar que o fluxograma neste exemplo foi associado a um algoritmo matemático. Na Figura 11 há uma possível representação de um fluxograma para o Algoritmo *Bubble Sort*, mas não se traduz na única possibilidade.

Analisou-se no Capítulo I que o tema fluxogramas é abordado na BNCC tanto na seção do Ensino Fundamental quanto na do Ensino Médio, sendo que na primeira há uma relação mais frequente com o significado de algoritmo voltado para Matemática e no Ensino Médio já se propõe um trabalho também com o aspecto computacional da definição de algoritmo.

Por fim, vale ressaltar que o estudo dos algoritmos deve estar atrelado ao estudo de fluxogramas, pois pela própria definição de algoritmo, tal relação é intrínseca, seja para reforçar conceitos e procedimentos matemáticos ou para estimular o pensamento computacional.

4 Sequência didática

A sequência proposta neste trabalho terá duração de 8 aulas, de 50 minutos cada, cabendo ao professor desenvolvê-la parcial ou completamente em momento que julgar propício e de acordo com a maturidade dos estudantes do 1º ao 3º ano do Ensino Médio.

O objetivo é desenvolver as atividades de modo a abordar os conteúdos matemáticos concernentes aos Algoritmos de Ordenação, desenvolver gradualmente os conceitos ligados ao tema, bem como as competências e habilidades matemáticas da BNCC.

Algumas atividades são propostas para serem desenvolvidas em grupos de estudantes e para tanto sugere-se as lições do livro *Planejando o Trabalho em Grupo- Estratégias para salas de aula heterogêneas*, das autoras Elizabeth G. Cohen e Rachel A. Lothan, Penso Editora, 3ª edição, RS, 2017. Contudo fica a critério de cada professor a forma como deve direcionar as atividades em grupo, mas nada impede que tais atividades possam ser vivenciadas e direcionadas individualmente.

As aulas da sequência didática foram organizadas de acordo como se apresenta na tabela a seguir:

Tabela 8 – Roteiro das aulas da Sequência Didática

Aula 1	Conhecendo e manipulando algoritmos.
Aula 2	Leitura e Produção de fluxogramas de algoritmos.
Aula 3	Criando um Algoritmo de Ordenação.
Aula 4	Conhecendo o *Bubble Sort*, *Selection Sort* e *Quick Sort*.
Aula 5	Ordenando com os algoritmos *Bubble Sort*, *Selection Sort* e *Quick Sort*.
Aula 6	Complexidade dos algoritmos *Bubble Sort*, *Selection Sort* e *Quick Sort*.
Aula 7	Comparando as complexidades dos algoritmos de ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort*.
Aula 8	Comparando graficamente as complexidades dos Algoritmos de Ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort*.

Fonte:Produzido pelo autor

Buscou-se, na sequência das aulas, direcionar as atividades de modo que haja uma relação com as Competências e Habilidades que constam na BNCC que seguem listadas adiante. Estão numeradas de acordo com a ordem em que aparecem no documento oficial de modo a facilitar sua localização na BNCC. É importante destacar que,

> Na (re)elaboração dos currículos e das propostas pedagógicas, é possível adotar outras organizações, recorrendo tanto às habilidades definidas nesta BNCC quanto a outras que sejam necessárias e que contemplem especificidades e demandas próprias dos sistemas de ensino e das escolas. (BRASIL, 2018, p. 542)

Verifica-se uma flexibilidade plausível disponibilizada pela BNCC, que oportuniza a criação de habilidades pontuais que abarquem demandas próprias do projeto da escola. Seguindo tais orientações, tomou-se aqui a liberdade de sugerir-se habilidades na sequência didática proposta.

Competência Específica 1 da BNCC

Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das ciências da Natureza e Humanas, ou ainda questões econômicas ou tecnológicas, divulgadas por diferentes meios, de modo a consolidar uma formação científica geral.

Habilidades

Não há na BNCC, para esta Competência Específica, uma habilidade correspondente ao tema.

Competência Específica 3 da BNCC

Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentos consistentes.

Habilidades

(02)Resolver e elaborar problemas cujos modelos são funções polinomiais do 1º e 2ª graus, em contextos diversos, incluindo ou não tecnologias digitais.

(05)Resolver e elaborar problemas com funções exponenciais e ou logarítmicas nos quais é necessário compreender e interpretar a variação das grandezas envolvidas, em contextos diversos.

(15)Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema.

Competência Específica 4 da BNCC

Compreender e utilizar com flexibilidade e precisão diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional, etc.) na busca de solução e comunicação de resultados de problemas.

Habilidades

(05)Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

(Sugerida pelo autor) Converter representações gráficas ou algorítmicas para representação algébrica na busca de solução de um problema.

Competência Específica 5 da BNCC

Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando recursos e estratégias como observações de padrões, experimentações e tecnologias digitais, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas.

Habilidades

(02)Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 1º grau ou do 2º grau do tipo $y = a \cdot x^2$.

Apresenta-se, na próxima tabela, uma síntese da distribuição das Competências e Habilidades nas respectivas aulas.

Tabela 9 – Aulas da sequência com respectivas competências e habilidades relacionadas

Aula	Competência	Habilidade
01	C03	H15
	C04	H05
02	C03	H15
03	C03	H15
	C04	H05
04	C03	H15
	C04	H05
05	C03	H15
	C04	H05
06	C01	Não há.
	C03	H02 e H05
	C04	H05 e H(sugerida)
	C05	H02
07	C01	Não há.
	C03	H02 e H05
	C04	H05 H(sugerida)
	C05	H02

Aula	Competência	Habilidade
08	C03	H02 e H05
	C05	H02

Vale salientar e exemplificar, para que não haja divergências, que a Habilidade 05 da Competência 03 difere da Habilidade 05 da Competência 04, ou seja, cada Habilidade está relacionada exclusivamente com uma única Competência.

1ª aula: Conhecendo e manipulando algoritmos.

Procedimentos

O professor irá explicar aos estudantes da maneira que achar mais conveniente o conceito de algoritmo tanto do ponto de vista matemático quanto computacional. Abordará exemplos por meio de algoritmos clássicos da matemática tais como: Algoritmo da divisão de números naturais com um divisor formado por um dígito e Algoritmo de Euclides para obter o m.d.c de dois números naturais. Poderá também exemplificar um algoritmo computacional escrito em linguagem natural ou algorítmica.
Após as devidas abordagens sobre o conceito de algoritmo o professor dividirá a turma em equipes sendo que cada equipe será subdividida em duas partes que executarão a seguinte atividade:
Uma das partes de cada equipe escreve um algoritmo que pode ser criação do grupo ou pesquisado e a outra parte deve executá-lo. Algumas sugestões que o professor pode disponibilizar para os estudantes são: algoritmo para divisão de dois números naturais sendo o divisor composto por dois dígitos e algoritmo para obter o m.m.c de dois números naturais.

2ª aula: Leitura e Produção de Fluxogramas de Algoritmos.

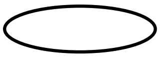
Procedimentos

O professor deve expor, por meio de slides, cartazes, etc, alguns exemplos de fluxogramas enfatizando como os comandos são realizados e sua função algorítmica. Em seguida, distribui fichas como a que segue abaixo para que os estudantes, em equipe, expressem por meio de fluxogramas algumas atividades cotidianas ou procedimentos matemáticos. É importante que o professor oriente os estudantes quanto à importância de escrever os comandos de forma clara e direta, sem que haja ambiguidade de interpretação. Uma sugestão para ajudar nessa tarefa é a apresentação do vídeo: <https://www.youtube.

com/watch?v=pdhqwbUWf4U&has_verified=1> (5:28 min), que mostra um pai que pede aos filhos para escreverem os comandos para que ele faça um sanduíche. À medida que os filhos apresentam os comandos e o pai vai executando é exposto os erros e as consequências de comandos ambíguos na escrita de algoritmos.

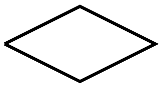
Ficha 1

Fluxograma que é um diagrama, escrito em uma notação gráfica simples, usado para representação visual de um algoritmo. Num fluxograma os principais símbolos são:

 Indica o início e o fim do algoritmo;

 Indica ação ou processo;

 Indica a direção do fluxo;

 Indica que uma decisão será tomada.

1) Represente numa cartolina, por meio de um fluxograma, cada atividade a seguir e exponha suas representações para as demais equipes.
Atividade 1: Assar um ovo.
Atividade 2: Tomar banho.
Atividade 3: Resolver uma equação do 2º grau.

3ª aula: Criando um Algoritmo de Ordenação.

Procedimentos

A turma será dividida em equipes e cada uma receberá a Ficha 2 com os comandos para desenvolvimento e registro das atividades juntamente com quatro fichas onde constam letras do alfabeto ou números. Cada equipe deverá criar uma estratégia de ordenação das fichas. O professor deve enfatizar que a estratégia deve ser um processo repetitivo, ou seja, será utilizado para colocar cada valor na posição correta sempre da mesma maneira. Após cada equipe criar sua própria estratégia de ordenação, tal estratégia deve ser comunicada e discutida com os demais estudantes.

Ficha 2

1) Cada equipe está recebendo cinco cartas com números ou letras diferentes. Como por exemplo: $\boxed{\text{D}}$ $\boxed{\text{M}}$ $\boxed{\text{T}}$ $\boxed{\text{A}}$. Escreva uma estratégia para ordenar tais elementos.

4ª aula: Conhecendo o *Bubble Sort*, *Selection Sort* e o *Quick Sort*.

Procedimentos

O professor deve fazer uma exposição por meio de cartazes, slides, vídeos, etc, do que é um Algoritmo de Ordenação e apresentar o *Bubble Sort*, *Selection Sort* e o *Quick Sort*. Uma sugestão é mostrar os vídeos:
<https://www.youtube.com/watch?v=lyZQPjUT5B4> (5:15 min)
<https://www.youtube.com/watch?v=Ns4TPTC8whw> (7:06 min)
<https://www.youtube.com/watch?v=ywWBy6J5gz8> (6:54 min)

Os vídeos sugeridos foram criados na Universidade de Sapienta na Romênia em cooperação com Mures Art Ensemble, uma espécie de fundação criada para guardar e desenvolver a cultura no país. Apresentam uma dança folclórica húngara com dançarinos numerados distintamente, por meio da coreografia os dançarinos executam as estratégias dos Algoritmos de Ordenação. É uma alternativa para se mostrar de forma divertida e atrativa um procedimento algorítmico.

Outra alternativa seria o professor dividir a turma em duas equipes para uma disputa, cada equipe teria que desenvolver o procedimento do algoritmo de forma correta, haverá em cada equipe um observador, que seria uma espécie de juiz. Caso o procedimento seja executado de forma errada, o observador pede para que se reinicie o procedimento. Ganha quem executar primeiro a ordenação correta com a estratégia do algoritmo.

O professor pode buscar outras situações que sirvam ao mesmo propósito dos vídeos e jogo sugeridos. Após a exposição, o professor poderá realizar indagações do tipo:
Qual algoritmo vocês acharam mais simples? E menos simples?
Vocês utilizaram alguma dessas estratégias na aula anterior?
Qual deles você acredita que leva mais tempo para ser executado?

Deverá observar e discutir com os alunos o funcionamento, isto é, a estratégia utilizada em cada Algoritmo de Ordenação e fazer as devidas intervenções no intuito de elucidar eventuais dúvidas.

5ª aula: Ordenando com os Algoritmos ***Bubble Sort***, ***Selection Sort*** e ***Quick Sort***.

Procedimentos

A turma será dividida em equipes (podem ser as mesmas da Aula 3) que receberão a Ficha 3 (ver modelo sugerido) e cinco fichas ou cartas de baralho numeradas. As fichas deverão ser ordenadas de três formas diferentes, utilizando as estratégias do *Bubble Sort*, *Selection Sort* e *Quick Sort* respectivamente, registrando todo o procedimento. O professor pede para cada equipe socializar suas respostas e comparar com as estratégias que foram criadas pelos estudantes na aula anterior.

Ficha 3

1) Cada equipe está recebendo cinco cartas com valores diferentes. Como por exemplo: $\boxed{4}$ $\boxed{-2}$ $\boxed{0}$ $\boxed{8}$ $\boxed{5}$. Estas cartas deverão ser ordenadas utilizando os Algoritmos de Ordenação a seguir e registrar todo o processo em um material a parte.

Ordenando com o *Bubble Sort*: Se o objetivo é ordenar os valores em forma decrescente, então, a posição atual é comparada com a próxima posição e, se a posição atual for maior que a posição posterior, é realizada a troca dos valores nessa posição. Caso contrário, não é realizada a troca, apenas passa-se para o próximo par de comparações.

Ordenando com o *Selection Sort*: A ordenação por seleção ou *Selection Sort* consiste em selecionar, por comparação, o menor item e colocar na primeira posição, selecionar o segundo menor item e colocar na segunda posição, segue estes passos até que reste um único elemento.

Ordenando com o *Quick Sort*: O vetor é dividido em duas partes a partir de um elemento denominado pivô, antes do pivô são colocados os elementos menores ou iguais ao mesmo e depois dele os elementos maiores. O processo se repete em cada uma das partes (subvetores) sucessivamente até que se obtenha partes com apenas um elemento. Para finalizar, os resultados são combinados e obtém-se o resultado esperado.

6ª aula: Complexidade dos Algoritmos ***Bubble Sort***, ***Selection Sort*** e ***Quick Sort***.

Procedimentos

Novamente em equipes, os alunos receberão uma ficha com tabelas que serão preenchidas com a ajuda do professor, onde os mesmos devem escrever as comparações realizadas pelos algoritmos: *Bubble Sort*, *Selection Sort* e *Quick Sort*, como indicadas nas tabelas 2, 4 e 6, por exemplos. Haverá um esforço para escrever uma função matemática que expresse o total de comparações, ou seja, o trabalho realizado por cada Algoritmo de Ordenação estudado. No caso do *Quick Sort*, o professor pode recorrer também a árvore de distribuição binária como na Figura 8.

Ficha 4

1) Com o auxílio das tabelas a seguir, descubra uma função que represente o desempenho dos Algoritmos de Ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort*.

Bubble Sort: Vetor para ordenar (6, 12, 2, 7, 1)

VETOR	**Nº de comparações**
6 - 12 - 2 - 7 - 1	
⋮	⋮

Selection Sort: Vetor para ordenar (6, 12, 2, 7, 1)

VETOR	**Nº de comparações**
6 - 12 - 2 - 7 - 1	
⋮	⋮

Para o *Quick Sort*, pode-se utilizar como auxílio uma árvore de distribuição binária.

Quick Sort: Vetor para ordenar (6, 12, 2, 7, 1)

Níveis(altura)	**Nº de subproblemas**	Tamanho do subproblema
⋮	⋮	⋮

7ªaula: Comparando as complexidades dos algoritmos de ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort*.

Procedimentos

Novamente em equipes, os alunos receberão a ficha 5 incluindo tabelas que serão preenchidas com a ajuda do professor, onde os mesmos devem calcular o número de comparações realizadas pelos algoritmos: *Bubble Sort*, *Selection Sort* e *Quick Sort*, como indicadas na tabela 7, por exemplo. Pode haver necessidade do uso de calculadora científica para obtenção dos resultados. O professor pode usar tabelas comparativas das complexidades dos algoritmos no melhor caso, caso médio e também pior caso. Feito isso, o professor poderá fazer alguns questionamentos tais como:

- Para um vetor com 16 elementos, qual estratégia é mais eficiente?

- Para um vetor com 512 elementos, qual o algoritmo é mais eficiente?
-Para um conjunto de 32.768 elementos qual algoritmo executa o menor números de comparações?
-Qual dos algoritmos você escolheria para ordenar uma lista com 10 elementos? E com 500 elementos? E com 10.000 elementos?

Ficha 5

1) Com o auxílio das tabelas a seguir, calcule o desempenho dos Algoritmos de Ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort* em cada caso e realize comparações.

Melhor caso

Nº de elementos	***Bubble Sort***: $O(n)$	***Selection Sort***: $O(n^2)$	***Quick Sort***: $O(n \cdot \log_2 n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Pior caso

Nº de elementos	***Bubble Sort***: $O(n^2)$	***Selection Sort***: $O(n^2)$	***Quick Sort***: $O(n^2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Caso médio

Nº de elementos	***Bubble Sort***: $O(n^2)$	***Selection Sort***: $O(n^2)$	***Quick Sort***: $O(n \cdot \log_2 n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

8ª aula: Comparando graficamente as complexidades dos algoritmos de ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort*.

Procedimentos

O professor deverá realizar uma exposição com orientações sobre Complexidade de Algoritmos de Ordenação, enfatizando e exemplificando a representação assintótica nos melhores e piores casos. Os estudantes serão divididos em equipes nos mesmos moldes das aulas anteriores e receberão a Ficha 6 com os comandos para representarem graficamente os desempenhos dos Algoritmos de Ordenação *Bubble Sort*, *Selection Sort* e *Quick Sort* nos melhores casos.

Ficha 6

1) Utilize a tabela para calcular o desempenho dos Algoritmos de Ordenação no melhor caso.

Nº de elementos	***Bubble Sort***: $O(n)$	***Selection Sort***: $O(n^2)$	***Quick Sort***: $O(n \cdot \log_2 n)$
8			
16			
64			
128			

2) Agora, utilizando os resultados da tabela, com o auxílio de uma malha quadriculada, represente graficamente o desempenho dos algoritmos.

Conclusão

Nesta pesquisa buscou-se inicialmente mostrar a relação da matemática com os avanços tecnológicos e da importância de se inserir, de forma apropriada, o estudo dos Algoritmos de Ordenação na Educação Básica, voltando-se principalmente para o Ensino Médio. Observamos que a abordagem de tal tema, no processo ensino-aprendizagem em Matemática, proporciona o desenvolvimento de competências e habilidades específicas sugeridas pela BNCC. Não se fez presente em nosso objetivo principal analisar a estrutura computacional de um Algoritmo de Ordenação, focamos apenas na estratégia utilizada e na análise matemática do desempenho.

No decorrer da análise dos Parâmetros Curriculares e BNCC pôde-se identificar vários conteúdos, competências e habilidades matemáticas relacionadas ao estudo dos Algoritmos de Ordenação, e confirmou-se a viabilidade de se abordar no Ensino Médio o tema aqui proposto. Contudo, apenas na BNCC apresentou-se, de forma explícita, o tema Algoritmos.

Quanto ao conteúdo matemático utilizado para que a análise dos Algoritmos de Ordenação sugeridos seja realizada de uma maneira correta e sem perda de rigor, verificou-se que o tema favorece o estudo de Séries Aritméticas no cálculo do número de comparações do *Bubble Sort* e *Selection Sort*, bem como o estudo das Funções Afim e Quadrática. Também se mostra uma boa oportunidade para revisar e aprofundar os conceitos e propriedades dos Logaritmos, da Função Logarítmica e Séries Geométricas na análise do *Quick Sort*.

No que se refere ao estudo do funcionamento e do desempenho dos Algoritmos de Ordenação, notou-se uma simplicidade nos Algoritmos *Bubble Sort* e *Selection Sort*, mas tanto o funcionamento quanto o cálculo do número de comparações do *Quick Sort* requerem um auxílio do professor para que os estudantes possam compreender melhor suas características. O tema Notação Assintótica se mostra como algo novo para o Ensino Médio e um aprofundamento se faz desnecessário no processo de ensino e aprendizagem, necessitando apenas que seja exposto como se utiliza tal notação para que o estudante possa utilizá-la nas comparações do desempenho. O tema Fluxogramas, para representação de algoritmos, foi abordado sucintamente e observou-se uma relativa dificuldade de utilizá-los para representar um Algoritmo de Ordenação, sendo seu uso mais indicado com algoritmos que possuem relação com conceitos e procedimentos matemáticos.

A sequência didática se propõe a servir como mais um subsídio utilizado por professores do Ensino Médio no processo de ensino e aprendizagem. Nela, tentamos resgatar tanto os conteúdos conceituais quanto os procedimentais que sirvam para os alunos desenvolverem as competências e habilidades específicas de matemática sugeridas na BNCC. Quatro das cinco competências foram abordadas na sequência e fazemos menção às que envolvem ações de representar, comunicar, raciocinar e argumentar.

Levando-se em consideração as análises e observações, constatou-se que não existem propostas consistentes para que se utilizem os Algoritmos de Ordenação, com preceitos didáticos, no Ensino Médio e que o tema se restringe, principalmente, aos cursos de graduação da área de computação. Tal ausência de propostas pode ser justificada devido à recente mudança curricular com o Novo Ensino Médio e implementação da BNCC. Se faz presente a convicção que é possível a abordagem didática do tema nas aulas de matemática no Ensino Médio e que a Sequência Didática aqui proposta se apresenta como uma boa ferramenta a ser utilizada.

Por fim, fica proposto que mais pesquisadores aprofundem o tema e produzam mais materiais didáticos correlatos. Sugere-se que a Sequência Didática proposta, seja aplicada e que se possa avaliar seu impacto no processo de aprendizagem, as possíveis dificuldades conceituais e procedimentais na abordagem do tema.

Referências

BRASIL. *Parâmetros Curriculares Nacionais de Matemática*. Brasília: MEC, 1998. 148 p.

BRASIL. *Parâmetros Curriculares Nacionais do Ensino Médio (PCNEM)*. Brasília: MEC, 2000. 148 p.

BRASIL. *PCN_+ Ensino médio: Orientações educacionais complementares aos Parâmetros Curriculares Nacionais-Ciências da Natureza, Matemática e suas Tecnologias*. Brasília: MEC, 2002. 142 p.

BRASIL. *DCN, Diretrizes Curriculares da Educação Básica*. Brasília: MEC, 2013. 562 p.

BRASIL, M. d. E. *Base Nacional Curricular Comum*. Brasil: MEC, 2018.

CORMEN, T. H. *Algoritmos: Teoria e Prática, tradução da segunda edição, [americana]*. São Paulo: Editora Campus, 2002.

FERRARI, F.; CECHINEL, C. Introdução a algoritmos e programação. Bagé-RS, 2008.

LEONARDO, F. M. d. *Conecxões com a Matemática, vol.1*. São Paulo-SP: Editora Moderna, 2016.

LIMA, E. L. et al. *A matemática do ensino médio*. [S.l.]: SBM, 1997. v. 1.

PERNAMBUCO, C. *Parâmetros para a Educação Básica do Estado de Pernambuco*. Recife-PE: Governo de Pernambuco, 2012.

RIBEIRO, M. R. da C. *Grafos, Algoritmos e Programação*. 152 p. Dissertação (Mestrado) — UFRPE, Recife-PE, 2018.