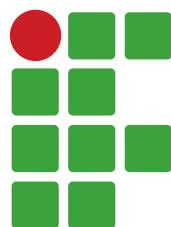


Instituto Federal de Educação, Ciência e Tecnologia de Brasília
Campus Taguatinga

Giovanni Lucas Oliveira da Silva

Linha de Pesquisa: Análise de Algoritmos e Complexidade de Computação

**Comparação de Algoritmos de Menor Caminho
em Grafos: Uma verificação prática da análise
assintótica de algoritmos sobre benchmarks de
grafos**



INSTITUTO FEDERAL
Brasília
Campus Taguatinga

Brasília, DF

2025

Instituto Federal de Educação, Ciência e Tecnologia de Brasília
Campus Taguatinga

Giovanni Lucas Oliveira da Silva

Área de Pesquisa: Análise de Algoritmos e Complexidade de Computação

**Comparação de Algoritmos de Menor Caminho em Grafos:
Uma verificação prática da análise assintótica de
algoritmos sobre benchmarks de grafos**

Documento apresentado como requisito parcial para matrícula na disciplina de Projeto de Conclusão de Curso do curso superior de Bacharelado em Ciência da Computação

Orientador: Daniel Saad Nogueira Nunes

Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Campus Taguatinga
Curso Superior de Bacharelado em Ciência da Computação

Brasília, DF

2025

Introdução

Dados, no ramo da ciência da computação, têm uma importância fundamental para o computador cumprir suas finalidades ([SANTOS, 2014](#)). Podemos usar os dados para representar coisas simples como números ou textos, ou coisas mais complexas como imagens e vídeos, ou até representações de pessoas, lugares e objetos reais. Toda aquisição de dados normalmente é feita com o propósito de ser utilizado para alguma finalidade, seja para o armazenamento de informações por si mesmo, ou para a realização de algum processamento e tomada de decisões.

Esses dados precisam ser armazenados de alguma forma eficiente e racional, para que possam ser analisados e utilizados posteriormente, e é aí que entram coisas como as estruturas de dados. Uma estrutura de dados é um modo de armazenar e organizar dados com o objetivo de facilitar acesso e modificações ([CORMEN et al., 2012](#)), existem várias estruturas de dados, cada uma com suas características, vantagens e desvantagens. A escolha da estrutura de dados correta para um determinado problema é crucial para o desempenho e eficiência do tratamento desses dados e, consequentemente, para a realização de tarefas e resolução de problemas.

Com a internet, a quantidade de dados gerados cresceu bastante, e os dados nela normalmente são representações de relações entre coisas, como pessoas, lugares e objetos, e essas relações podem ser representadas por uma estrutura muito importante chamada grafo. Um grafo é uma das estruturas de dados que mais é utilizada para representar essas relações pois ela é uma estrutura que relaciona esses objetos que nela chamamos de vértices, através de conexões que chamamos de arestas. Cada aresta ligando dois vértices representa uma relação entre esses dois vértices, como por exemplo, essa aresta pode representar uma amizade entre duas pessoas, ou uma rota entre duas cidades. Em aplicações do mundo real, grafos podem ser utilizados para representar as relações entre pessoas em redes sociais ou as diversas rotas entre cidades em aplicações de geolocalização. ([GRONER, 2019](#))

Tomando como por exemplo as aplicações de geolocalização, ter um grafo representando as rotas entre cidades torna esses dados mais fáceis de serem analisados e processados, se colocarmos valores nas arestas representando a distância entre as cidades, quando um usuário quiser ir de uma cidade A para uma outra cidade B, podemos com esse grafo analisar as diversas rotas possíveis entre essas duas cidades caso haja mais de uma rota, ou se até mesmo não houver uma rota direta entre essas duas cidades, e assim determinar diversas possibilidades de rotas que o usuário pode fazer para ir de A para B. Mas dentre todas essas rotas possíveis, surge problemas como, A e B são alcançáveis um do outro? Qual a rota mais curta entre A e B? Quais são as rotas que passam por uma outra cidade C? Esses são problemas entre muitos outros que podem surgir quando trabalhamos com essas questões, neste pré-projeto vamos focar em um dos problemas mais comuns e importantes que é o problema em determinar qual a rota mais curta entre dois vértices, esse

problema é mais referido como o problema do menor caminho.(LINTZMAYER; MOTA, 2023)(GRONER, 2019)

Esse problema é encontrado em diversas aplicações do mundo real, como nesse caso de geolocalização e navegação, em muitas outras aplicações como redes sociais, jogos, entre outros. E para solucionar esse problema, as pessoas desenvolveram soluções que chamamos de algoritmos, uma sequência de passos definidos que levam a uma solução.(CORMEN et al., 2012) Para o problema do menor caminho, existem diversos algoritmos desenvolvidos, cada um com suas características, vantagens e desvantagens.

Alguns dos algoritmos mais conhecidos para resolver o problema do menor caminho são o algoritmo de Dijkstra, o algoritmo de Bellman-Ford e o algoritmo de Floyd-Warshall, cada um desses algoritmos tem uma análise assintótica diferente, essas análises nos permitem prever o desempenho de um algoritmo sem a necessidade de implementá-lo e testá-lo, o que é muito útil quando estamos lidando com grandes quantidades de dados. (LINTZMAYER; MOTA, 2023) Porém, essas análises são apenas previsões teóricas, e o desempenho real de um algoritmo pode variar dependendo de muitos fatores. Por isso, é importante não apenas analisar os algoritmos teoricamente, mas também testá-los na prática.

Recentemente um novo algoritmo de menor caminho foi proposto em uma pesquisa intitulada 'Breaking the Sorting Barrier for Directed Single-Source Shortest Paths' (DUAN et al., 2025) este algoritmo tem um tempo de execução determinístico assintoticamente menor que o Dijkstra, o que é uma grande proposta, visto que o dijkstra é um dos algoritmos mais eficientes para resolver problemas de menor caminho em grafos. (LINTZMAYER; MOTA, 2023) Mesmo que a análise assintótica seja um bom indicador do desempenho, ela está interessada em ordem de crescimento superiores, isso é ela despreza constantes e ordens inferiores, que para implementações práticas, podem fazer com que algoritmos potencialmente mais rápidos se tornem mais lentos quando implementados e usados em situações reais.

Teóricamente esse algoritmo quebrou uma barreira que a muito tempo era considerada a mais eficiente nesse problema, mas como falado anteriormente, o desempenho real desse algoritmo será diferente do desempenho teórico, e com isso podemos indagar se esse algoritmo vai trazer ganhos práticos em relação às práticas atuais, será que é de fato mais rápido que o Dijkstra em grafos reais? É para responder essa questão que se propõe esse pré-projeto.

Proposta

O trabalho se propõe a investigar a discrepância entre o ganho de eficiência teórico e prático do novo algoritmo para o Problema do Menor Caminho de Fonte Única (SSSP),

em comparação com o algoritmo de Dijkstra. Para isso, o cerne deste projeto foca na realização de um estudo de benchmark.

Primeiramente, será realizada a implementação detalhada de ambos os algoritmos, o novo algoritmo assintoticamente mais rápido e o algoritmo de Dijkstra. Em seguida, o foco será direcionado à avaliação prática por meio de um ambiente de testes realista. Utilizaremos conjuntos de dados e frameworks de benchmark de grafos. Esses recursos permitirão a geração de grafos sintéticos e a simulação de cenários que variam em escala, densidade e estrutura, refletindo as complexidades encontradas em aplicações do mundo real.

O objetivo final é quantificar e analisar o tempo de execução de cada algoritmo nessas diferentes situações em grafos. Dessa forma, será possível determinar se o avanço assintótico do novo algoritmo se traduz em ganhos de desempenho prático significativos, e em quais condições específica ele consegue, de fato, superar a eficiência consolidada do algoritmo de Dijkstra.

Justificativa

O problema do menor caminho não é um problema puramente acadêmico, ele é um problema que é presente em vários ambientes descritos anteriormente, e por isso, ter algoritmos mais eficientes para resolver esse problema é de grande importância. E para que possamos ter certa clareza sobre a eficiência de algoritmos que se propõem a serem mais rápidos que os já utilizados atualmente, é necessário que seja verificada a sua validade prática, ou seja, criando evidências empíricas que comprovem a sua superioridade. Esse estudo baseado em benchmarks permite o fornecimento de dados concretos sobre o desempenho real, aplicando os fatores ignorados pela notação assintótica em seu resultado.

Esses resultados terão impacto prático direto pois, se o novo algoritmo se mostrar de fato mais eficiente, essa informação influenciará decisões de engenharia e desenvolvimento de software, levando a escolhas mais informadas sobre quais algoritmos utilizar em aplicações que envolvem o problema do menor caminho. Além disso, esse estudo contribuirá para a comunidade acadêmica e profissional ao fornecer uma análise crítica e detalhada sobre a aplicabilidade prática de avanços teóricos em algoritmos, incentivando futuras pesquisas e inovações na área.

Proposta Metodológica Preliminar

Para atingir esses objetivos, a proposta metodológica é baseada em uma pesquisa experimental e quantitativa. Ambos os algoritmos serão implementados fielmente na linguagem de programação C++ para garantir a máxima eficiência e controle sobre o

overhead. O *benchmark* será realizado em um ambiente controlado, utilizando ferramentas como o *BGGS* (*Big Graph Generator Suite*) para gerar grafos sintéticos com variação de escala e densidade. O tempo de execução será mensurado com alta precisão e repetido múltiplas vezes. Por fim, os resultados serão analisados para comparar o desempenho dos algoritmos em diferentes cenários, validando a aplicabilidade prática do avanço teórico do algoritmo. (Duan et al., 2025).

Referências

- CORMEN, T. H. et al. *Algoritmos: Teoria e Prática*. 3. ed. Rio de Janeiro: Elsevier, 2012. Tradução da 3^a edição de *Introduction to Algorithms*. Citado 2 vezes nas páginas [2](#) e [3](#).
- DUAN, R. et al. Breaking the sorting barrier for directed single-source shortest paths. *arXiv preprint arXiv:2504.17033*, July 2025. Versão de 31 de julho de 2025. Disponível em: [⟨https://arxiv.org/abs/2504.17033v2⟩](https://arxiv.org/abs/2504.17033v2). Citado 2 vezes nas páginas [3](#) e [5](#).
- GRONER, L. *Estruturas de dados e algoritmos com JavaScript*. 2. ed. São Paulo: Novatec Editora, 2019. Tradução da obra *Learning JavaScript Data Structures and Algorithms - Third Edition* (Packt Publishing, 2018). Tradução de Lúcia A. Kinoshita. Revisão gramatical de Tássia Carvalho. ISBN 978-85-7522-728-2. Disponível em: [⟨https://www.kufunda.net/publicdocs/Estruturas%20de%20dados%20e%20algoritmos%20com%20JavaScript%20\(Loiane%20Groner\).pdf⟩](https://www.kufunda.net/publicdocs/Estruturas%20de%20dados%20e%20algoritmos%20com%20JavaScript%20(Loiane%20Groner).pdf). Citado 2 vezes nas páginas [2](#) e [3](#).
- LINTZMAYER, C. N.; MOTA, G. O. *Análise de Algoritmos e de Estruturas de Dados*. [S.l.], 2023. Versão de 4 de agosto de 2023. Disponível em: [⟨https://www.ime.usp.br/~mota/bookFiles/livro_AAED.pdf⟩](https://www.ime.usp.br/~mota/bookFiles/livro_AAED.pdf). Citado na página [3](#).
- SANTOS, A. C. dos. *Algoritmo e Estrutura de Dados I*. [S.l.], 2014. Texto da disciplina produzido em 2008 e revisado em 2010, 2012 e 2014. Licenciado sob Creative Commons Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. Disponível em: [⟨https://educapes.capes.gov.br/bitstream/capes/176522/2/TextoAEDI_SI_UFAL_AiltonCruz.pdf⟩](https://educapes.capes.gov.br/bitstream/capes/176522/2/TextoAEDI_SI_UFAL_AiltonCruz.pdf). Citado na página [2](#).