

# ANÁLISE COMPARATIVA DE ALGORITMOS DE BUSCA – PROBLEMA DO LABIRINTO

João Pedro Neves de Siqueira  
Email: joao.siqueira1@utp.edu.br  
Universidade Tuiuti do Paraná

## Resumo

Este trabalho tem como objetivo comparar, de forma prática, diferentes algoritmos de busca aplicados à resolução de labirintos. Foram testados quatro algoritmos bem conhecidos: Busca em Largura (BFS), Busca em Profundidade (DFS), A\* (A-estrela) e Busca Gulosa. Para entender o desempenho de cada um, levei em conta três aspectos principais: o tempo que cada algoritmo leva para encontrar a saída, o quanto de memória ele consome durante o processo e quantos passos são necessários até chegar ao objetivo. As comparações foram feitas com base em seis labirintos diferentes. No fim das contas, percebi que nenhum algoritmo é o melhor em tudo — cada um tem seus pontos fortes e fracos, e a escolha depende muito do que se prioriza: velocidade, economia de recursos ou um caminho mais eficiente.

**Palavras-chave:** algoritmos de busca, labirinto, BFS, DFS, A\*, busca gulosa.

## Abstract

This work presents a practical comparison between different search algorithms applied to the maze-solving problem. The algorithms tested were Breadth-First Search (BFS), Depth-First Search (DFS), A\* (A-star), and Greedy Search. The evaluation focused on three main aspects: how fast each algorithm finds the exit, how much memory it uses during the process, and how many steps it takes to reach the goal. The comparison was based on six different maze instances. In the end, the results showed that no single algorithm is the best at everything — each one has its strengths and weaknesses, and the ideal choice depends on what you're prioritizing: speed, resource efficiency, or path quality.

## 1 Introdução

Problemas de busca em grafos estão presentes em diversas aplicações da computação, como jogos, inteligência artificial e análise de dados.

Neste contexto, labirintos são uma excelente representação visual e prática para aplicar algoritmos de busca, uma vez que permitem observar o comportamento de cada algoritmo frente a diferentes estruturas de caminhos.

Este trabalho visa comparar quatro algoritmos clássicos: BFS, DFS, A\* e Gulosa, considerando diferentes instâncias de labirintos, com métricas de tempo, memória e quantidade de passos.

## 2 Metodologia

Foi desenvolvido um sistema em Python para aplicar os quatro algoritmos em seis diferentes labirintos representados por arquivos texto (maze1.txt a maze6.txt).

Cada algoritmo percorre o labirinto do ponto inicial até o final, retornando três métricas principais:

- Tempo de execução (em segundos);
- Uso de memória (em KB);

- Passos (número de movimentos do início até o fim).

Cada script foi executado individualmente para cada instância, e os dados foram coletados diretamente do terminal.

### Observação Pessoal

Notei que, ao rodar os testes mais de uma vez, o número relacionado ao uso de memória variava um pouco mesmo sem mudar nada no código. Isso acontece por causa do jeito que o sistema operacional e o Python gerenciam a memória a cada execução.

Então, os valores que coloquei nas tabelas foram tirados de uma única execução para cada labirinto, mas é bom ter em mente que eles podem mudar levemente se rodar de novo. Pra ter resultados mais precisos, o ideal seria rodar várias vezes e tirar uma média.

## 3 Resultados

Comparativo entre os algoritmos de busca por tempo, memória e passos:

Tabela 1: Resultados dos algoritmos em cada labirinto

Lab	Algoritmo	Tempo (s)	Memória (KB)	Passos
1	BFS	0.000000	12.00	32
	DFS	0.000000	0.00	32
	A*	0.000000	8.00	32
	Gulosa	0.000997	4.00	32
2	BFS	0.000000	12.00	74
	DFS	0.000997	0.00	74
	A*	0.000000	4.00	74
	Gulosa	0.001025	0.00	74
3	BFS	0.000998	24.00	146
	DFS	0.000000	8.00	146
	A*	0.001001	8.00	146
	Gulosa	0.000000	4.00	146
4	BFS	0.000996	20.00	128
	DFS	0.001027	0.00	128
	A*	0.001968	8.00	128
	Gulosa	0.000995	0.00	128
5	BFS	0.001925	44.00	500
	DFS	0.001080	24.00	500
	A*	0.001975	80.00	500
	Gulosa	0.002933	0.00	500
6	BFS	0.001995	60.00	408
	DFS	0.000995	12.00	408
	A*	0.001975	32.00	408
	Gulosa	0.001000	0.00	408

## 4 Análise dos Resultados

- **Tempo:** DFS foi o algoritmo que, em geral, apresentou os menores tempos, seguido de perto pela Gulosa. Entretanto, A\* e BFS se mantiveram estáveis com tempos competitivos, principalmente em labirintos menores.
- **Memória:** DFS e Gulosa se destacaram por usarem menos memória. A\* tende a usar mais, especialmente em instâncias maiores. Isso se deve ao uso de filas de prioridade e listas de caminhos.

- **Passos:** Todos os algoritmos chegaram ao mesmo número de passos por instância, indicando que, apesar das abordagens diferentes, todos encontraram soluções igualmente eficientes em termos de trajeto.

## 5 Conclusão

Depois de testar e comparar todos os algoritmos, deu pra perceber que cada um tem seus pontos fortes e fracos, dependendo do que a gente está buscando.

Se a prioridade for economizar memória, por exemplo, a Busca em Profundidade pode ser uma boa escolha. Já se a ideia for ter um controle melhor do tempo e da direção da busca, a Busca Gulosa pode funcionar bem.

O A\* foi o que mais se destacou no equilíbrio entre velocidade e qualidade do caminho encontrado — é um algoritmo mais inteligente e, na maioria das vezes, entregou bons resultados.

O BFS também foi eficiente, principalmente na garantia de encontrar o menor caminho, mas com um custo maior em termos de memória.

No fim das contas, não existe um algoritmo “melhor” pra tudo. A escolha ideal vai depender muito do tipo de problema, do tamanho do labirinto e do que se espera da solução: mais velocidade, menos consumo de memória ou o caminho mais curto possível.