

# Relatório final dos algoritmos de busca.

Arthur Siqueira e Silva, Ihan Belmonte Bender

## Uniform Cost Search:

O algoritmo expande sempre o nodo com menor custo em relação ao inicial, realizando uma exploração radial. Como não usa uma heurística, acaba expandindo nodos desnecessários, de forma cega.

## Testes:

*Mapa: bigTeste (feito pelo aluno).*  
*Nodos expandidos: 628*  
*Tempo de execução: 0.1 segundos*  
*Score: 438*

*Mapa: bigMaze*  
*Nodos expandidos: 620*  
*Tempo de execução: 0.1 segundos*  
*Score: 300*

*Mapa: mediumMaze*  
*Nodos expandidos: 269*  
*Tempo de execução: 0.0 segundos*  
*Score: 442*

*Mapa: tinyMaze*  
*Nodos expandidos: 15*  
*Tempo de execução: 0.0 segundos*  
*Score: 502*

## Hill Climbing Search:

O algoritmo não alcança o resultado esperado a maioria das vezes, pois fica preso em um máximo global.

## Testes:

*Mapa: bigTeste (feito pelo aluno).*

*Nodos expandidos: 9*

*Tempo de execução: 0.0 segundos*

*Score: - infinito (não chega ao resultado final)*

*Mapa: bigMaze*

*Nodos expandidos: 3*

*Tempo de execução: 0.0 segundos*

*Score: - infinito (não chega ao resultado final)*

*Mapa: mediumMaze*

*Nodos expandidos: 9*

*Tempo de execução: 0.0 segundos*

*Score: - infinito (não chega ao resultado final)*

*Mapa: tinyMaze*

*Nodos expandidos: 9*

*Tempo de execução: 0.0 segundos*

*Score: 502 (Consegue encontrar o máximo global)*

## Simulated Annealing Search (Têmpera Simulada):

O algoritmo não é totalmente determinista, já que se baseia em probabilidades aleatórias, que são usadas para fugir de possíveis máximos locais e ocasionalmente ficar próximo do máximo global. A cada interação, a chance de tomar uma decisão ruim diminui, fazendo com que o algoritmo tenda ao Hill Climbing Search.

Foi definido o máximo de iterações como 10000.

Testamos 2 vezes em cada mapa para se ter ideia da aleatoriedade presente.

Observa-se que as pontuações são baixas, visto que demoram para encontrar o máximo global (quando encontram).

## Testes:

*Mapa: bigTeste (feito pelo aluno).*

*Nodos expandidos: 10000*

*Tempo de execução: 0.2 segundos*

*Score: -8264*

*Mapa: bigTeste (feito pelo aluno)*

*Nodos Expandidos: 10000*

*Tempo de execução: 0.2 segundos*

*Score: - infinito (não chegou)*

*Mapa: bigMaze*  
*Nodos expandidos: 10000*  
*Tempo de execução: 0.2 segundos*  
*Score: - infinito (não chegou)*

*Mapa: mediumMaze*  
*Nodos expandidos: 10000*  
*Tempo de execução: 0.2 segundos*  
*Score: - infinito*

*Mapa: tinyMaze*  
*Nodos expandidos: 10000*  
*Tempo de execução: 0.2 segundos*  
*Score: 472*

*Mapa: bigMaze*  
*Nodos expandidos: 10000*  
*Tempo de execução: 0.2 segundos*  
*Score: - infinito (não chegou)*

*Mapa: mediumMaze*  
*Nodos expandidos: 10000*  
*Tempo de execução: 0.2 segundos*  
*Score: - 8340*

*Mapa: tinyMaze*  
*Nodos expandidos: 10000*  
*Tempo de execução: 0.2 segundos*  
*Score: 178*

## A\* Search:

O algoritmo encontra sempre o melhor caminho, além de ser pouco custoso, explorando menos nodos que a maioria dos outros algoritmos de busca.

Rodamos testes com duas heurísticas: manhattan e euclidean, e concluiu-se que no problema dos labirintos, a manhattan se sai melhor, explorando menos nodos, como era de se esperar devido ao formato e das possibilidades de movimento.

## Testes: Usando Manhattan Heuristic

*Mapa: bigTeste (feito pelo aluno).*  
*Nodos expandidos: 279*  
*Tempo de execução: 0.0 segundos*  
*Score: 438*

*Mapa: bigMaze*  
*Nodos expandidos: 549*  
*Tempo de execução: 0.0 segundos*  
*Score: 300*

*Mapa: mediumMaze*  
*Nodos expandidos: 221*  
*Tempo de execução: 0.0 segundos*  
*Score: 442*

*Mapa: tinyMaze*  
*Nodos expandidos: 14*  
*Tempo de execução: 0.0 segundos*  
*Score: 502*

## Testes: Usando Euclidean Heuristic

*Mapa: bigTeste (feito pelo aluno).*  
*Nodos expandidos: 441*  
*Tempo de execução: 0.0 segundos*  
*Score: 438*

*Mapa: bigMaze*  
*Nodos expandidos: 557*  
*Tempo de execução: 0.0 segundos*  
*Score: 300*

*Mapa: mediumMaze*  
*Nodos expandidos: 226*  
*Tempo de execução: 0.0 segundos*  
*Score: 442*

*Mapa: tinyMaze*  
*Nodos expandidos: 13*  
*Tempo de execução: 0.0 segundos*  
*Score: 502*