

## Segunda avaliação de GBC063 - AARE Etapa I / 2020

### Trabalho de implementação de algoritmos de busca

Este trabalho consiste em implementar na linguagem Python (v.3) os algoritmos descritos abaixo para simular o comportamento de um agente cruzando um labirinto.

### Entregas

Os trabalhos podem ser feitos em grupos de, no máximo 3 alunos, e as entregas consistem de:

- os códigos comentados referentes a todos os algoritmos.
- um relatório do experimento (conforme especificado abaixo).

As entregas devem ser realizadas via plataforma MS Teams.

As entregas serão aceitas até às 24:00 do dia 01/09.

### Sobre os programas suporte disponibilizados

Os programas bases disponibilizados permitem criar objetos em Python que representam labirintos aleatórios 2D a partir do algoritmo de Kruskal e apresentam uma interface simples para explorar esses labirintos. [referência <https://github.com/138paulmiller/PyMaze> ]

Esses programas foram ligeiramente modificados, de modo a simplificar os labirintos e a interface para o teste de algoritmos de busca.

Três bases/diretórios são disponibilizados na pasta 'Materiais de classe' de 'Arquivos' de 'Geral':

- Visual – *onde os movimentos (passo-a-passo) do agente dentro do labirinto podem ser visualizados (indicado para fazer debug).*
- OnlyTracking - *onde o labirinto não é visto - somente os passos do agente (indicado para fazer debug).*
- NoVisual - *onde não há interface gráfica (somente o resultado da busca é apresentado em txt).*

Os códigos podem ser modificados para, por exemplo, se obter somente os valores da solução ou dos passos. O importante é que os algoritmos desenvolvidos não recebam (do labirinto) mais informação do que sua posição atual e as possíveis posições subsequentes.

## Algoritmos

Os códigos comentados referentes aos algoritmos listados abaixo devem ser implementados em Python (v.3) e, depois de testados, entregues em formato texto (com tabulação pronta para teste).

- Busca em profundidade
- Busca por aprofundamento iterativo
- Usando a distância de Manhattan do estado atual até o objetivo como função heurística,
  - Busca por descida de encosta
  - Busca por Têmpera simulada (Simulated Annealing) para duas funções de variação de temperatura (ao longo do tempo) distintas - de preferência uma  $T(t)$  que funcione bem para o problema.

## Relatório

Um documento contendo introdução, resultados e conclusão deve ser entregue em pdf.

Na seção Resultados devem ser apresentadas tabelas e gráficos do tipo boxplot para os seguintes experimentos:

Experimento 1 – Teste dos algoritmos de busca sem informação para 100 labirintos 10x10

Busca	Movimentos realizados na busca	Tamanho do caminho da solução
Por profundidade	<média dos 100 valores>	<média dos 100 valores>
Por Aprofundamento iterativo	<média dos 100 valores>	<média dos 100 valores>

Experimento 2 – Teste dos algoritmos de busca que usam a distância de Manhattan como informação para 100 labirintos 10x10 \*

Busca	Distância final até o objetivo	Movimentos realizados na busca	Tamanho do caminho da solução
Descida de Encosta	<média dos 100 valores>	<média dos 100 valores>	<média dos 100 valores>
Têmpera Simulada $T_1(t)$	<média dos 100 valores>	<média dos 100 valores>	<média dos 100 valores>
Têmpera Simulada $T_2(t)$	<média dos 100 valores>	<média dos 100 valores>	<média dos 100 valores>

\* OBS: de modo a representar as paredes do labirinto na tela, cada movimento é representado por uma mudança de 2 linhas/caracteres.