

Trabalho 1

1 - Descrição do trabalho

Este trabalho tem como objetivo a implementação dos algoritmos de busca vistos em aula e aplicá-los em um problema real de busca. O problema selecionado para este trabalho é o de encontrar um caminho entre um ponto de origem e um ponto destino em um grafo que modela caminhos entre locais.

1.1 - Implementação

Visando criar um ambiente propício de testes, serão utilizados neste trabalho os *grafos-knn*, pois possibilitam a geração de diferentes configurações e sua topologia final é semelhante a mapas. Os seguintes algoritmos de busca serão utilizados nos experimentos:

- Busca em profundidade
- Busca em largura
- Algoritmo Best-First
- Algoritmo A
- Algoritmo A*

O pseudo-código abaixo elucida a geração de um grafo-knn, portanto, sendo n o número de vértices no grafo e k o número de vizinhos (arestas) conectadas a cada vértice, temos que:

```
função gera_grafo_knn(n, k):  
    lista_vértices <- gera_vértices(n)  
    lista_arestas <- gera_arestas(lista_vertices, k)  
    retorna lista_vertices, lista_arestas
```

Onde:

- A função **gera_vertices** retorna uma lista de n vértices, onde para cada vértice v é associada as coordenadas x e y . As coordenadas x e y são dadas aleatoriamente, e devem respeitar os limites do eixo- x e eixo- y que devem ser limitados ao valor de n . Por exemplo, tendo $n = 1000$, as coordenadas de x e y devem estar entre os valores de 0 e 1000. Portanto, quanto maior o valor de n , maior o tamanho do plano xy gerado, o que aumenta as possibilidades de distribuição dos vértices de acordo com o tamanho do grafo.
- A função **gera_arestas** recebe a *lista_vértices* e o valor de k . Para cada vértice origem v_o são conectadas k arestas, onde cada vértice destino v_d é selecionado aleatoriamente. O valor (peso) da aresta é dado pela distância geométrica entre v_o e v_d .

Assim, ao final da função **gera_grafo_knn** é gerado um grafo não-direcionado com n vértices e grau médio igual a k .

1.1 - Experimentos

Os seguintes experimentos são propostos para este projeto:

- I. Dada um grafo-knn, selecionar um vértice origem e um vértice destino e aplicar os algoritmos de busca selecionados para este projeto. Para cada algoritmo, mostrar o caminho final percorrido (cada vértice que faz parte do caminho solução) e a distância final percorrida.
- II. Gerar os seguintes grafos-knn: (5000, 3), (5000, 5) e (5000, 7). Aleatoriamente selecionar, no mínimo, 20 pares de vértices (origem, destino) e aplicar os algoritmos de busca com objetivo de reportar a distância média percorrida por cada algoritmo, além do tempo médio.
- III. Realizar uma comparação entre o algoritmo A* e o algoritmo de Djisktra.

A realização de experimentos complementares, como por exemplo adicionar outras configurações de grafos-knn no experimento II, terão avaliação extra. Assim, como a implementação de meios de visualização que facilitem a verificação dos nós visitados pelos algoritmos.

Ao fim, um relatório, de no máximo 10 páginas, deve ser entregue. Neste relatório deve conter: introdução, descrição da implementação dos algoritmos de busca, resultados dos experimentos propostos e uma discussão sobre os resultados obtidos. O trabalho deve ser apresentado em sala de aula, por meio de slides, e deve conter os resultados obtidos nos experimentos.

2 - Datas

- **Entrega do relatório e código-fonte** até o final do dia **17 de outubro**.
- **As apresentações ocorrerão nos dias 18, 21 e 25 de outubro**. O dia e a ordem das apresentações devem ser preenchidos na seguintes planilha:
<https://docs.google.com/spreadsheets/d/1BWz5o8GUWMYPLmaFDjGxmPwFuBcl4BptcSdb3H9XTps/edit?usp=sharing>

3 - Informações importantes

- O grupo deve ter 4 integrantes;
- O projeto pode ser desenvolvido em qualquer linguagem de programação;
- Relatório (em .pdf) e código-fonte do projeto devem ser entregues no Tidia (escaninho) por apenas um integrante do grupo.
- O **tempo de apresentação** será de **15 minutos**.
- **Trabalhos plagiados receberão nota 0.**