

# TF - Trabalho Final

## Objetivo

O objetivo deste trabalho é praticar a implementação de algoritmos utilizando grafos e a criação de um algoritmo para esta estrutura.

## Tarefas Obrigatórias

Ler um arquivo texto com a descrição de um grafo, com o seguinte formato:

```
vertices  
# x y  
arestas  
a b custo
```

Cada vértice e aresta está descrito em uma lista. Os atributos dos vértices (no arquivo) são identificador (número inteiro) e as coordenadas X e Y do vértice (double). Os atributos das arestas são dois identificadores de vértices (inteiros) e um custo para acessar a aresta (double).

A distância entre os vértices não está incluída no custo e deve ser calculada de acordo com a posição dos dois vértices que conectam as arestas.

Você deve implementar o Algoritmo de Dijkstra para calcular a menor distância entre dois pontos, levando em consideração o custo e a distância entre os pontos.

Com o algoritmo implementado, devem ser implementados os seguintes algoritmos:

- Criar um algoritmo para encontrar uma boa rota entre vários pontos. Lembre-se que este é o problema do Caixeiro Viajante, cuja solução ou é impossível de obter em um tempo razoável, ou é bastante complexa para implementar. Aqui, é pedido que criem um algoritmo simples, que encontre uma rota entre os pontos desejados.
- Criar um algoritmo que determine pontos de paradas em um caminho, que atendam as necessidades de “reabastecimento” e “descanso”.

Para os cálculos leve em consideração o seguinte:

- Consumo de Combustível: 15 Km/l
- Autonomia do veículo: 600 Km.
- Velocidade média de viagem: 80Km/h
- Tempo máximo de direção sem paradas: 3h

O trabalho deve ser feito em grupos de 4 pessoas.

Na penúltima aula do semestre, deve ser realizada uma apresentação do trabalho desenvolvido, mostrando a organização da equipe para a implementação do trabalho, explicando o sistema, e os dois algoritmos criados (Caixeiro Viajante e Paradas no Caminho).

## Data de Entrega

Penúltima aula do semestre.

## Dicas

Consultem códigos disponíveis na internet. Evitem copiá-los (pode ser mais difícil adaptá-los do que entender e implementar o algoritmo).

Consulte a explicação da Wikipedia:

- [http://en.wikipedia.org/wiki/Dijkstra's\\_algorithm](http://en.wikipedia.org/wiki/Dijkstra's_algorithm) (Inglês, melhor)
- [http://pt.wikipedia.org/wiki/Algoritmo\\_de\\_Dijkstra](http://pt.wikipedia.org/wiki/Algoritmo_de_Dijkstra) (Português)

## Objetivos de Aprendizado

Desenvolver um algoritmo utilizando uma estrutura de dados complexa.

## Avaliação

Será avaliada a conclusão das tarefas obrigatórias do trabalho.

Conceitos melhores podem ser obtidos demonstrando qualidade no código, nos registros de *commit*, e na documentação do código.

## Pontos Extras

Existem várias possibilidades de melhorias neste trabalho. A partir das tarefas obrigatórias, todo o trabalho extra será considerado.

## Observações

Utilize o arquivo em anexo para testar o seu sistema.