

Trabalho Prático 1

Neste trabalho prático, você deve implementar duas heurísticas simples para a resolução do problema do Caixeiro Viajante. Neste problema você deve encontrar o menor caminho existente que passe por todos os vértices (sem repeti-los) e volte ao vértice de partida.

As duas heurísticas que você deve implementar são:

- **Força Bruta:** Consiste em resolver o problema testando todas as possibilidades possíveis e no final escolher a melhor. Ou seja, o ciclo com o menor custo. Apesar de sempre encontrar a solução ótima, esta heurística é extremamente cara computacionalmente e rapidamente se torna inviável de ser computada já com poucos vértices.
- **Vizinho Mais Próximo:** Também conhecida como heurística gulosa, essa estratégia visa obter ganhos imediatos a cada iteração. Ou seja, se o objetivo é encontrar o caminho com menor custo, durante o processo de execução, o algoritmo sempre escolhe o vértice adjacente do vértice atual com o menor custo. Apesar de ter um custo computacional linear, esta heurística raramente encontra a solução ótima.

Como sua aplicação deve funcionar

Você deve implementar uma aplicação para cada heurística. Ou seja, você deve criar dois diretórios, um para cada projeto, com os seguintes nomes: **forca_bruta** e **guloso**. Sua aplicação deve receber dois parâmetros. O primeiro parâmetro informa a localização do arquivo de entrada contendo as informações do grafo. O segundo parâmetro é o nome do arquivo de saída que sua aplicação irá gerar contendo o caminho encontrado pela heurística e qual o tamanho dele. Exemplo:

```
forca_bruta.exe arquivo_entrada.txt saida_forca_bruta.txt
```

ou:

```
guloso.exe arquivo_entrada.txt saida_guloso.txt
```

OBS: O comando funciona de maneira similar ao executar com `dotnet run`.

Formato do arquivo de entrada

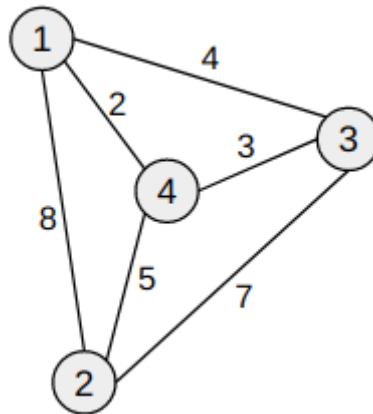
O arquivo de entrada contém uma matriz quadrada representando a distância de cada vértice para todos os demais. Veja o exemplo abaixo para um grafo com 4 vértices:

```
0 8 4 2
8 0 7 5
4 7 0 3
2 5 3 0
```

arquivo_entrada.txt

Cada linha do arquivo representa a distância de um vértice até os demais. Por exemplo, a distância do vértice V3 até o vértice V2 é igual a 7.

O grafo abaixo é uma ilustração do grafo definido pelo arquivo acima:



Formato do arquivo de saída

O arquivo de saída que sua aplicação irá gerar deve conter o ciclo encontrado pela heurística, começando e terminando no vértice V1. Cada vértice do ciclo deve estar em uma linha do arquivo. Além do ciclo, a última linha do arquivo deve conter o tamanho total do ciclo. Abaixo um exemplo do menor caminho encontrado pela heurística de força bruta aplicada ao grafo acima:

```
1
3
2
4
1
18
```

saida_forca_bruta.txt

OBS: Os vértices no arquivo de saída devem começar do 1 e não do zero.

O que deve ser entregue

Além dos arquivos com o código-fonte da sua aplicação escrita em C#, **você deve entregar um documento em PDF, com duas páginas no máximo, discutindo como você implementou as heurísticas**. Portanto, segue a lista dos arquivos esperados na entrega do trabalho:

