
Aluno:	Matrícula:	Valor: 25,0	Nota:
--------	------------	-------------	-------

Trabalho I

1. Objetivos.

- Desenvolver a habilidade de programação de algoritmos em grafos.
- Reforçar o aprendizado sobre os algoritmos para o problema do caminho mínimo.
- Avaliar o impacto da complexidade computacional no tempo de execução.

2. Descrição.

O trabalho consiste em implementar algoritmos para resolver problemas de caminho mínimo em grafos. O projeto Grafos, disponibilizado no GitHub, deve ser adaptado com a implementação dos seguintes algoritmos para caminho mínimo: Busca Largura (para grafos não ponderados), Dijkstra e Bellman-Ford (para grafos com arestas de peso negativo). Um conjunto de grafos de teste foi disponibilizado para avaliar os trabalhos. O seu programa deve, dadas as características do grafo de entrada, escolher automaticamente o algoritmo mais eficiente para resolvê-lo e a melhor estrutura de dados para representá-lo.

Adicionalmente, deve-se gravar um vídeo de até 10 minutos descrevendo detalhadamente sua implementação e os resultados observados para um conjunto de grafos de teste. Sugiro o uso da ferramenta Google Meet para tal. De modo a assegurar a autoria do vídeo, é solicitado que o mesmo seja gravado com a câmera ligada mostrando o(s) autor(es) do trabalho.

O seu programa deve solicitar ao usuário o nome do arquivo contendo o grafo (no formato Dimacs) e os vértices de origem e de destino s e t . Como saída, seu programa deve informar a sequência de vértices do caminho mínimo de s a t , bem como o custo do caminho e o tempo de execução (em segundos) do algoritmo. O tempo deve ser contado a partir do ponto em que foram informados os vértices origem e destino até a exibição da resposta. Ex.:

```
Informe o arquivo: <toy.txt>
```

```
Origem: <0>
```

```
Destino: <3>
```

```
Processando...
```

```
Caminho: [0, 1, 2, 3]
```

```
Custo: 5
```

```
Tempo: 0.003s
```

3. Teste dos algoritmos.

Cada trabalho deverá validar sua implementação sobre os conjuntos de dados disponíveis no arquivo “Datasets.zip”. Esse conjunto de grafos são provenientes de aplicações reais, tais como locomoção em grandes cidades e conexões do Facebook. A tabela abaixo apresenta uma breve descrição sobre cada grafo de teste:

Arquivo	Descrição
toy.txt	Exemplo dado em aula, que pode ser checado manualmente.
rg300_4730.txt	Grafo aleatório com 300 vértices e 4730 arestas.
rome99c.txt	Descreve um mapa das estradas de Roma em 1999.
facebook_combined.txt	Conexões entre contatos do Facebook (com 4039 usuários).
USA-road-dt.DC.txt	Descreve um mapa das estradas de Washington DC.
USA-road-dt.NY.txt	Descreve um mapa das estradas de New York.
web-Google.txt	Mapa da Web coletado pelo Google em 2002.

Sugiro utilizar a instância “toy” para testar a corretude dos algoritmos visto que é fácil verificar seu resultado manualmente.

4. Avaliação.

O trabalho deverá ser feito individualmente ou em dupla e enviado via Moodle (código (preferencialmente link para o GitHub) e vídeo) até as 23:59h do dia 16/05/22. Os algoritmos serão avaliados com relação à corretude das implementações e saídas obtidas e com relação à eficiência (tempo de execução). O professor executará os programas recebidos e avaliará os tempos de execução observados para avaliar a corretude e a eficiência dos programas. Pontos extras serão concedidos ao trabalho com a melhor média dos tempos de execução. Caso se tenha alguma dúvida com relação à autoria do trabalho o professor poderá solicitar uma apresentação presencial ao aluno (ou dupla).

Bom trabalho!