Universidade Federal de Ouro Preto – UFOP
Departamento de Computação e Sistemas – DECSI
CSI466 – Teoria dos Grafos (2024-1)
Prof. Samuel Souza Brito Data: 16/08/24

Primeiro Trabalho Prático

1 Objetivos

- Desenvolver habilidades de programação de algoritmos em grafos.
- Reforçar o aprendizado sobre os algoritmos de caminhos mínimos em grafos.
- Avaliar o impacto da complexidade computacional dos algoritmos nos tempos de execução.

2 Descrição

Conforme visto em sala de aula, existem diferentes algoritmos para resolver o problema do caminho mínimo. A escolha de um algoritmo irá depender, por exemplo, da estrutura do problema a ser resolvido, da eficiência a ser obtida no pior caso ou no caso médio e da quantidade de memória a ser utilizada. Este trabalho consiste em implementar os algoritmos de *Dijkstra*, *Bellman-Ford* e *Floyd-Warshall* para encontrar o caminho mínimo entre dois vértices em um grafo direcionado e ponderado. Os algoritmos deverão ser implementados em Python e incorporados ao projeto contido no repositório TeoriaDosGrafos-2024-1 do GitHub, desenvolvido durante as aulas de Teoria dos Grafos.

Seu programa deve receber os seguintes parâmetros via linha de comando: o nome do arquivo contendo o grafo, o vértice de origem e o vértice de destino. O arquivo a ser lido pelo seu programa deve estar no formato DIMACS (ver Aula 03 - Representação Computacional). Como saída, seu programa deve informar, para cada algoritmo, a sequência de vértices contidos no caminho mínimo entre o vértice de origem e o vértice de destino, bem como o custo do caminho e o tempo de execução (em segundos). O tempo deve ser contabilizado a partir do momento em que o algoritmo é executado até a exibição da resposta. Um exemplo de execução e de saída esperada do seu programa é ilustrado a seguir, considerando o grafo armazenado no arquivo "toy.txt", vértice de origem 0 e vértice de destino 3.

```
PS C:\Users\samuel\csi466\tp1> python main.py toy.txt 0 3
Processando...
------------------------------------------------------------
Algoritmo de Dijkstra:
Caminho mínimo: [0, 2, 1, 3]
Custo: 5
Tempo: 0.003s
------------------------------------------------------------
Algoritmo de Bellman-Ford:
Caminho mínimo: [0, 2, 1, 3]
Custo: 5
Tempo: 0.018s
------------------------------------------------------------
Algoritmo de Floyd-Warshall:
Caminho mínimo: [0, 2, 1, 3]
Custo: 5
Tempo: 0.123s
------------------------------------------------------------
```

3 Experimentos Computacionais

Os algoritmos deverão ser validados utilizando os grafos disponíveis no arquivo "grafos.zip". Esses grafos são provenientes de aplicações reais, tais como locomoção em grandes cidades e conexões entre perfis do Facebook. A tabela abaixo apresenta uma breve descrição sobre cada grafo de teste:

Arquivo	Descrição
toy.txt	Grafo pequeno para ser checado manualmente.
facebook_combined.txt	Conexões entre perfis do Facebook (com 4039 perfis).
rg300_4730.txt	Grafo aleatório com 300 vértices e 4730 arestas.
rome99c.txt	Mapa das estradas de Roma em 1999.
USA-road-dt.DC.txt	Mapa das estradas de Washington DC.

4 Entrega e Avaliação

Este trabalho prático deve ser realizado em grupos de **até três alunos**. Deverão ser entregues os códigos-fonte devidamente comentados e um relatório. O relatório deve conter detalhes sobre as implementações, experimentos computacionais, análise dos resultados e principais conclusões. Informações sobre como executar seu programa também devem constar no relatório. Os experimentos devem calcular, para cada algoritmo e grafo, a média aritmética do tempo de execução e do custo do caminho mínimo considerando 10 execuções, variando os vértices de origem e destino. Dessa forma, a seguinte tabela deve ser preenchida e inserida na seção dos experimentos computacionais do relatório:

Grafo	**Dijkstra**		**Bellman-Ford**		**Floyd-Warshall**	
	T.médio(s)	**Custo médio**	**T.médio(s)**	**Custo médio**	**T.médio(s)**	**Custo médio**
toy						
facebook_combined						
rg300_4730						
rome99c						
USA-road-dt.DC						

Caso a execução não finalize em até 600 segundos, marque *TEMPO_LIMITE* na tabela. Caso não tenha sido possível carregar o grafo em memória, marque *MEMORIA_EXCEDIDA*.

O relatório e o código-fonte devem ser enviados em um **único arquivo compactado via Moodle até as 23:59 horas do dia 15/09/24**. Somente um integrante do grupo deve realizar a entrega.

O valor deste trabalho prático é **20,0 pontos**. As implementações serão avaliadas com relação à corretude e à eficiência (tempo de execução). Em caso de dúvidas sobre a autoria do trabalho, o professor poderá solicitar uma apresentação presencial aos alunos.

ATENÇÃO: Todas implementações devem utilizar as estruturas (matriz ou lista de adjacências) e os métodos implementados no repositório *TeoriaDosGrafos-2024-1* do GitHub. Implementações que não utilizem essas estruturas e métodos receberão nota zero. Consultas a documentações, manuais e tutoriais são válidas, desde que não haja cópia e as fontes sejam devidamente citadas. Em caso de plágio ou detecção de uso de Inteligência Artificial para gerar os códigos, todos os alunos envolvidos receberão nota zero. Não serão aceitos trabalhos entregues fora do prazo.