

Estudo Dirigido - Otimização em Redes (BCC463)

Marco Antonio M. Carvalho
Universidade Federal de Ouro Preto
Departamento de Computação

24 de março de 2025

Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outras fontes, ou que não puder ser explicado, acarretará na perda da nota;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não serão aceitos;
- Não serão considerados algoritmos parcialmente implementados.

1 Algoritmo de Johnson

O algoritmo de Johnson, proposto em 1977, determina os caminhos mais curtos entre todos os pares de vértices em um grafo direcionado com arcos ponderados. É permitido que alguns dos pesos dos arcos sejam negativos, mas não pode haver nenhum ciclo de peso negativo. O algoritmo funciona usando o algoritmo Bellman-Ford para calcular uma transformação do grafo de entrada que remove todos os pesos negativos, permitindo que o algoritmo de Dijkstra seja usado no grafo transformado. A implementação deve reutilizar os códigos desenvolvidos anteriormente.

Especificação da Entrada

A primeira linha da entrada contém dois inteiros n , m , indicando a quantidade de vértices e a quantidade de arcos. Em seguida haverá m linhas, cada uma contendo três inteiros, indicando o vértice de origem (enumerados de 1 a n), o vértice de destino e o peso dos arcos, que podem ser negativos.

Especificação da Saída

Após executar o algoritmo, imprima a matriz de distâncias entre todos os n vértices. Cada linha da saída deve ser terminada com '\n'.

Exemplo de Entrada

```
6 7
1 2 -5
2 3 -7
2 5 2
3 6 -3
4 1 -6
5 4 50
6 5 1
```

Exemplo de Saída

```
0 -5 -12 36 -14 -15
35 0 -7 41 -9 -10
42 37 0 48 -2 -3
-6 -11 -18 0 -20 -21
44 39 32 50 0 29
45 40 33 51 1 0
```

Estrutura do código

O código-fonte deve ser organizado em um único arquivo, nomeado *principal.c*. A separação das operações em funções e procedimentos está a cargo do aluno, porém, não deve haver acúmulo de operações dentro uma mesma função/procedimento.

Diretivas de Compilação

```
$ gcc principal.c -c
$ gcc principal.o -o programa
```