Estudo Dirigido - Parte 1 - Teoria dos Grafos (BCC204)

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

16 de agosto de 2021

Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Na plataforma run.codes, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação do run.codes considera o tempo de execução e o percentual de respostas corretas;
- 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 outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.
- Serão realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- A avaliação do estudo dirigido consiste de: (I) avaliação do run.codes; (II) verificação de plágio; (III) verificação de aderência ao enunciado e à estrutura obrigatória dos códigos; e (IV) entrevista. As quatro etapas da avaliação são eliminatórias.

1 Busca em Largura

De modo geral, pode-se dizer que o projeto de bons algoritmos para a determinação de estruturas ou propriedades dos grafos depende do domínio de técnicas que permitam examinar com eficiência vértices e arestas. A esse tipo de procedimento denomina-se, genericamente, "busca em grafos" ou "percurso em grafos".

Neste estudo dirigido é pedido ao aluno que implemente um dos algoritmos de percurso mais populares para grafos: a **Busca em Largura**. A implementação deve utilizar uma **lista de adjacências** como representação computacional.

Especificação da Entrada

A primeira linha da entrada contém quatro inteiros n, m b e i, indicando a quantidade de vértices, a quantidade de arestas/arcos, um valor binário indicando se o grafo é direcionado (valor 1) ou não (valor 0) e um índice do vértice (enumerados de 1 a n) a partir do qual será realizada a busca.

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 das arestas/arcos.

Especificação da Saída

Após realizar a busca, indique a ordem de visitação dos vértices a partir do vértice i, um vértice por linha. Por padronização, os vizinhos de um vértice qualquer devem ser explorados em **ordem crescente de índice**. No caso de grafos desconectados, **não é necessário** reiniciar a busca. Cada linha da saída deve ser terminada com '\n'.

Exemplo de Entrada

```
4 6 0 1
1 2 1
1 3 1
1 4 1
2 3 1
2 4 1
3 4 1
```

Exemplo de Saída

```
1
2
3
4
```

Estrutura do código

O código-fonte deve ser modularizado corretamente em três arquivos: principal.(c, cpp), BFS.(h, hpp) e BFS.(c, cp). O arquivo principal.(c, cpp) deve apenas invocar as funções e procedimentos definidos no arquivo BFS.(h, hpp). 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 C

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

Diretivas de Compilação C++

```
$ g++ BFS.cpp -c
$ g++ principal.cpp -c
$ g++ BFS.o principal.o -o programa
```