Estudo Dirigido - Parte 2 - Teoria dos Grafos (BCC204)

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

7 de março de 2022

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 Algoritmo de Ford Fulkerson

O algoritmo de Ford-Fulkerson (assim designado em honra de Lester Randolph Ford, Jr e Delbert Ray Fulkerson) é um algoritmo utilizado para resolver problemas de fluxo em rede. O algoritmo é empregado quando se deseja encontrar um fluxo de valor máximo que faça o melhor uso possível das capacidades disponíveis na rede em questão.

O problema resolvido pelo algoritmo é o de encontrar um fluxo máximo em uma rede. Uma rede pode ser uma rede elétrica, um sistema de transporte de fluidos ou a distribuição de produtos ao longo de uma rede de transportes, como uma malha ferroviária ou rodoviária. Por exemplo, deseja-se transportar o máximo de minério de ferro através de uma rede ferroviária, limitadas pela capacidade de cada via. O tratamento aqui dado ao algoritmo supõe a existência de um único "ponto de entrada" (uma fonte) e de um único "ponto de saída" (um sumidouro).

Na segunda parte do estudo dirigido é pedido ao aluno que implemente este algoritmo. A busca em profundidade desenvolvida na primeira parte do estudo dirigido deve ser utilizada obrigatoriamente na implementação do algoritmo de Ford Fulkerson. Por padronização da saída, os vértices devem ser analisados obrigatoriamente em **ordem crescente de índice**.

Especificação da Entrada

A primeira linha da entrada contém dois inteiros n e m, indicando a quantidade de vértices e a quantidade de arestas/arcos. O vértice 1 sempre será a fonte e o vértice n sempre será o sumidouro. Adicionalmente, o grafo sempre será direcionado.

Em seguida haverá m linhas, cada uma contendo três inteiros, indicando o vértice de origem do arco (enumerado de 1 a n), o vértice de destino do arco (enumerados de 1 a n) e o limite superior para o fluxo no referido arco. O limite inferior para o fluxo em qualquer arco será sempre zero.

Especificação da Saída

Para cada caminho de aumento de fluxo encontrado pelo algoritmo, deve ser apresentada uma linha com os rótulos de cada vértice no mesmo formato das notas de aula, i.e., [x, $\pm$,ξ_y], em que x indica o vértice a partir do qual o vértice atual y foi rotulado, $\pm$ indica rotulação a partir de um arco direto (+) ou reverso (-) e ξ_y indica o quanto o fluxo pode ser aumentado no caminho de s até o vértice atual y.

Cada linha da saída deve ser terminada com '\n'.

Exemplo de Entrada

```
6 8
1 2 2
1 3 4
2 3 4
2 5 1
3 4 2
4 5 6
4 6 5
5 6 10
```

Exemplo de Saída

```
[1, +, 2] [2, +, 2] [3, +, 2] [4, +, 2] [5, +, 2]
[1, +, 4] [3, -, 2] [2, +, 1] [5, -, 1] [4, +, 1]
```

Estrutura do código

O código-fonte deve ser modularizado corretamente em três arquivos: principal.(c, cpp), FordFulkerson.(h, hpp) e FordFulkerson.(c, cp). O arquivo principal.(c, cpp) deve apenas invocar as funções e procedimentos definidos no arquivo FordFulkerson.(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. A implementação deve se basear **obrigatoriamente** no código da busca em profundidade desenvolvido na parte anterior do estudo dirigido.

Diretivas de Compilação C

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

Diretivas de Compilação C++

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