

Prática 3 Prazo: 08/01/2018
Atividade em Dupla / 20% da 3ª Avaliação

1. DESCRIÇÃO

Entrada.

Seu programa deve ser capaz de ler um grafo de um arquivo texto. O formato do grafo no arquivo será o seguinte. A primeira linha informa o número de vértices do grafo. Cada linha subsequente informa as arestas. Um exemplo de um grafo e seu respectivo arquivo texto é dado na figura abaixo. Os vértices deverão ser alfanuméricos. Seu programa deverá permitir carregar grafos orientados e não orientados, e grafos ponderados e não ponderados.

Exemplo para grafos não orientados:

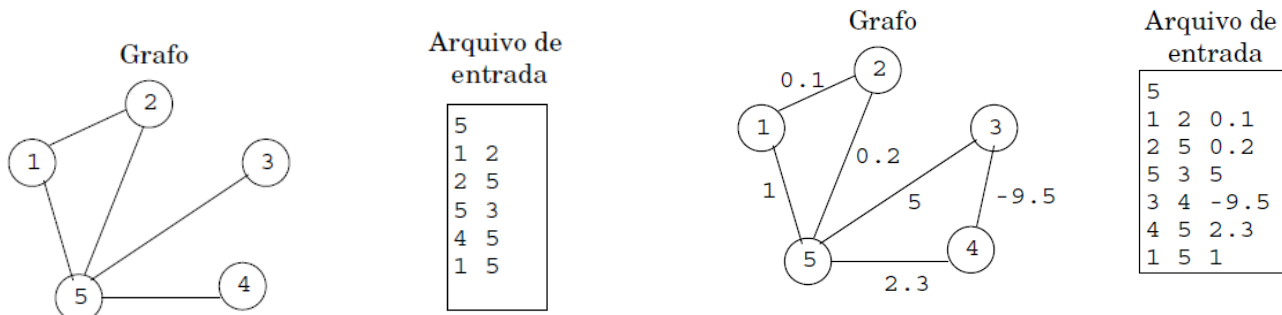


Figura 1: Exemplo de grafo com pesos e o formato do arquivo.

Para grafos orientados deverá receber a entrada no seguinte formato usando > para indicar sentido direito, < para esquerdo e <> para indicar que sentido duplo. Exemplo:

```
5
1 > 2
2 > 5
5 < 4
1 <> 5
```

Problema (0,5pt)

O Governo do Estado irá disponibilizar internet gratuitamente em todo o estado desta forma, deseja interligar as n estações de transmissão espalhadas pelas cidades do estado usando canais de comunicação. Cada par de estações tem uma capacidade de transmissão de internet diferente, que são previamente conhecidas. O Governo deseja escolher $n-1$ canais (o mínimo possível) de tal forma que todas as estações estejam conectadas pelos canais de transmissão e a capacidade de transmissão total seja máxima. Implemente um algoritmo eficiente para este problema. Considere o Grafo ponderado (V, E) , onde V é o conjunto de estações e E é o conjunto de canais entre as estações. Defina o peso $w(e)$ de uma aresta e e per E como a capacidade de transmissão do canal correspondente.

Saída

Seu programa deve ser capaz ainda de:

1. gerar um arquivo texto com as seguintes informações sobre o grafo: número de vértices, número de arestas; (0,1pt)

2. representar grafos utilizando tanto uma matriz de adjacência, quanto uma lista de adjacência. O usuário poderá escolher a representação a ser utilizada. (0,1pt)
3. Realizar busca em largura e profundidade. Seu programa deve ser capaz de percorrer o grafo utilizando busca em largura e busca em profundidade. (0,2) O vértice inicial será dado pelo usuário. A respectiva árvore de busca deve ser gerada assim como o nível de cada vértice na árvore (nível da raiz é zero). Estas informações devem ser fornecidas em arquivo ou console. Para descrever a árvore gerada, basta informar o pai de cada vértice e seu nível no arquivo de saída.
4. Apresentar os vértices de cada componente conectado.(0,1)
5. Retornar a mensagem se o gráfico é cíclico ou acíclico.(0,2)
6. Permitir que o usuário visualize a Árvore Geradora Mínima usando Prim ou Kruskal (0,3pt)
7. Apresentar o caminho mais curto entre dois vértices apresentados pelo usuário. (0,3pt)

2. Testes da implementação

Elabore vários casos de testes e apresente os resultados no relatório para cada teste realizado.

4. Entrega

- Código fonte do programas em C/C++, Java ou Python (bem indentado e comentado).
- Relatório dos testes realizados. (0,2pt)
- Upload no SIGAA.