

Prática 3 Prazo: 13/12/2019
Atividade Individual / 30% 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:

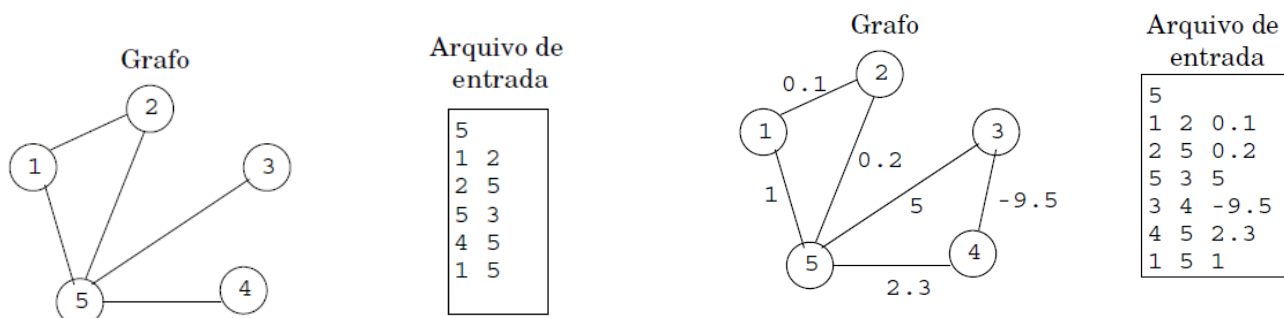


Figura 1: Exemplo de grafo com pesos e o formato do arquivo.

Faça as seguintes implementações:

- 1) Representar grafos utilizando tanto uma matriz de adjacência, quanto uma lista de adjacência. O usuário poderá escolher a representação a ser utilizada. (1 pt)
- 2) Implemente o algoritmo de busca em largura;
 - a. Implemente um método para retornar o número de arestas existentes entre o vértice de origem da BFS e um outro vértice destino. (1 pt)
- 3) Seu programa deve imprimir a lista de adjacências na ordem inversa de carregamento, no seguinte formato: (1pt)
<Nº de vértices> vértices; <Nº de arestas> arestas
1: 8 7 5
2:
3: 4 2
4: 3 1
.
.
- 4) A excentricidade de um vértice v é o comprimento do caminho mais curto desse vértice para o vértice mais distante de v . O diâmetro de um grafo é a excentricidade máxima de qualquer vértice. O raio de um grafo é a menor excentricidade de qualquer vértice.
 - a. Implemente um método para retornar a excentricidade de um vértice V (1.pt)

- b. Implemente um método para retornar o diâmetro de um grafo G (1.pt)
 - c. Implemente um método para retornar o raio de um grafo G (1.pt)
- 5) Implemente uma versão do algoritmo de PRIM e do KRUSKAL para retornar a floresta de abrangência mínima (MST) de um grafo ponderado que não está necessariamente conectado. (2.pt)
- 6) Você está ajudando na investigação de um caso. Para tanto, você precisa encontrar quais são as cidades que estão dentro de um determinado raio de distância fornecido pelo usuário. Logo, seu programa deverá receber o vértice de entrada e em seguida, deverá retornar os vértices que estão dentro do raio de distância fornecido. OBS: a distância está em Km. (2.pt)

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. (2 pt)
- Upload no SIGAA.