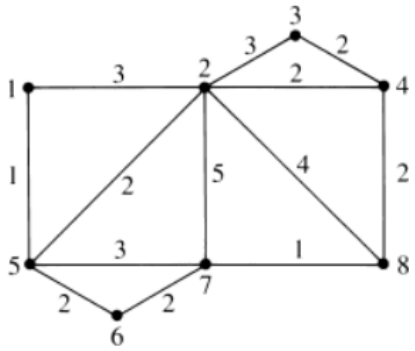
Algoritmos e Estruturas de Dados II

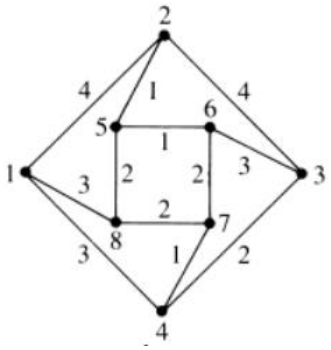
Exercícios – Árvores Geradoras Mínimas

1) Para cada um dos seguintes grafos, execute os algoritmos de Prim e Kruskal passo-a-passo e obtenha uma árvore geradora mínima:

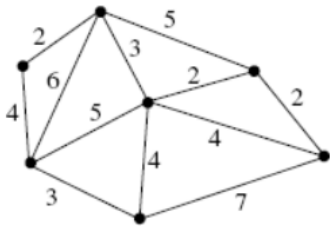
a)



b)



2) Uma cidade está planejando construir caminhos para bicicletas ligando diversos parques municipais. A figura a seguir mostra um mapa com as distâncias entre os parques. (Alguns caminhos diretos teriam que cruzar ruas de muito movimento, de modo que essas distâncias não constam do mapa.) Encontre quais os caminhos que devem ser feitos de modo a ligar todos os parques com um custo mínimo.



3) Como seria o algoritmo para obter uma árvore geradora máxima?

4) Como os algoritmos de Prim e Kruskal poderiam ser utilizados para a construção de labirintos? Realize uma pesquisa e sumarize os resultados.

5) Seja o seguinte enunciado (cujo código-fonte se encontra no arquivo ZIP no Moodle):

Foi fornecido um arquivo com um conjunto de pontos separados por ";" (arquivo CSV)

```
-24.1036;-49.7900
-19.5603;-46.9653
-9.8661;-56.1044
-22.6394;-50.4525
-10.2500;-59.3833
...
```

a) Construir um programa em Java para:

- Ler o conjunto de pontos para uma estrutura de dados adequada;
- Criar um grafo não dirigido valorado nas arestas com todos os pontos, gerando ligações entre cada ponto e os 3 mais próximos. Ou seja, o peso de cada aresta corresponde à distância Euclidiana entre os pontos. Esta pode ser calculada usando o Teorema de Pitágoras para dois pontos (x_1, y_1) e (x_2, y_2):

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

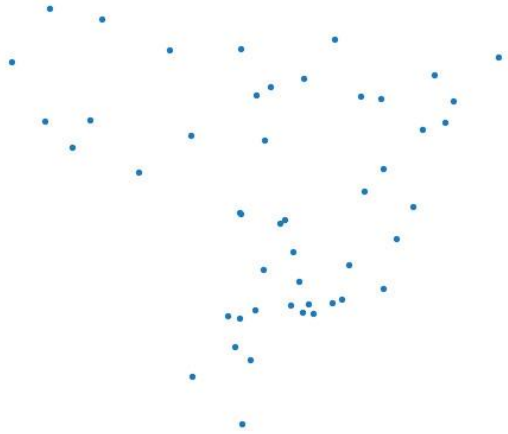
- A partir do grafo, implementar e executar um algoritmo de árvore geradora mínima (Prim ou Kruskal), gerando um arquivo com todas as arestas, no seguinte formato (indicar o número do vértice inicial, o número do vértice final e 1 se essa aresta pertence à árvore geradora mínima e 0 caso contrário):

```
0 10 1
0 20 0
0 30 0
0 40 0
10 50 1
0 52 1
0 55 1
50 58 1
20 22 0
30 32 0
32 28 0
58 99 1
...
```

b) Visualizar o resultado de saída:

- Através dor programa "plotgraph.py" (em Python, na linha de comando). Ele espera como parâmetro o arquivo de saída no formato apresentado anteriormente. Por exemplo:

```
$ python plotgraph.py saida.txt
```



- Alternativamente, gere a saída no formato .dot do programa GraphViz (no Linux) e visualize com o link https://dreampuf.github.io/GraphvizOnline/