

Trabalho 06 - Algoritmo de Prim

Data de entrega: 13/12/2020

Importante:

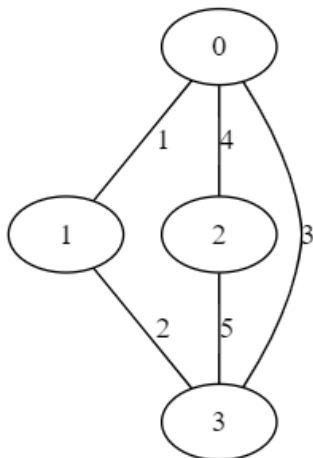
- Não olhe códigos de outros grupos ou da internet. Exceto o que é fornecido.
- TODOS os membros do grupo devem participar e compreender completamente a implementação.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes, onde o “Número de Matrícula” deverá ser o número do seu grupo.

Esse trabalho deverá ser realizado em grupo, com os grupos já definidos na disciplina. Neste trabalho o seu grupo deverá encontrar uma Árvore Geradora Mínima usando o Algoritmo de Prim.

Suponha o seguinte problema: Existe um conjunto de cidades e é preciso levar energia para todas elas. Em uma dessas cidades está a usina que produz energia. Para conectar um par de cidades u e v , temos um custo conhecido $c(u,v)$, caso não seja possível conectar diretamente u e v , podemos considerar que $c(u,v)$ é infinito. Esse problema é conhecido como Árvore Geradora Mínima, formalmente:

Dado um grafo não direcionado $G = (V, E)$, com custos $c(u,v)$ para cada aresta (u,v) em E , desejamos encontrar um subconjunto T de E , tal que $G' = (V, T)$ seja conexo, não contenha ciclos e tenha o menor custo possível.

Neste trabalho você deverá ler da entrada padrão do sistema um grafo no seguinte formato: dois inteiros representando o número n de vértice, e o número m de arestas. Depois m linhas representando cada uma das arestas, cada aresta contém 3 inteiros, com o índice dos vértices incidentes em cada aresta e um inteiro com o custo daquela aresta, os vértices estão indexados de 0 até $n-1$. Considere por exemplo o grafo abaixo com 4 vértices e 5 arestas e a entrada que o representa:



```
4 5
0 1 1
0 2 4
0 3 3
1 3 2
2 3 5
```

Você deverá implementar o Algoritmo de Prim começando pelo vértice 0, e deverá imprimir as arestas na ordem em que são adicionadas na árvore, o formato para impressão é “(u,v)” em que $u < v$, sem espaços e uma quebra de linha no final. E depois você deverá imprimir o custo da árvore. Para o exemplo de entrada acima sua saída deveria ser:

```
(0,1)
(1,3)
(0,2)
7
```

Normalmente o Algoritmo de Prim é implementado usando uma Fila de Prioridade, e o grafo representado por Lista de Adjacência, o que confere ao algoritmo sua eficiência. Porém neste trabalho você não precisa se preocupar com essa eficiência, então você poderá usar por exemplo uma Matriz de Adjacência e pode encontrar as arestas para adicionar da árvore de maneira mais trivial (procurando por todas por exemplo). Você receberá uma implementação parcial do algoritmo, cujo uso é opcional, que já contém o código leitura de entrada.

- Você deverá implementar seu código em linguagem C.
- Seu programa deve executar no run.codes em poucos segundos.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!