SCC0216 - Modelagem computacional em grafos
Professor: Alneu de Andrade Lopes
Estagiário PAE: Alan Valejo

LAB 3: MST - PRIM

1. Especificação

O laboratório consiste na implementação do algoritmo PRIM que encontra a árvore geradora mínima em grafos. Deverá ser utilizada a linguagem de programação C99.

2. Descrição da entrada

O programa principal deverá ler da entrada padrão a lista de arestas do grafo e representálo em memória – matriz ou lista de adjacência, conforme desejar. Em seguida, o programa deverá executar o algoritmo PRIM, encontrar a árvore gerado mínima e imprimir o resultado na saída padrão.

Na primeira linha da entrada, haverá o descritor do grafo contendo 2 números separados por espaço. Os números indicam, nesta ordem, o número de vértices e de arestas do grafo. Nas linhas seguintes, as arestas do grafo serão representadas por 3 números indicando os vértices de origem, destino e peso.

O programa deverá imprimir as arestas da MST em ordem de inclusão na árvore. Após imprimir a MST, insira uma quebra de linha.

3. Dicas

- O vetor de predecessor será utilizado como saída para árvore geradora mínima.
- O sistema considerará que a expansão dos nós são feitas por ordem do índice, portanto, se utilizarem lista de adjacência, mantenha a lista de vértices adjacentes ordenada.

4. Submissão

O exercício deverá ser entregue pelo sistema run.codes. Todos os alunos deverão submeter seus códigos no 'Exercício 4: MST - PRIM' até o final da aula. Somente a última submissão será considerada. Todas as demais submissões serão desconsideradas, incluindo aquelas dentro do período normal de submissão. Os exercícios deverão submetidos em um arquivo .zip contendo código-fonte do programa e um Makefile para compilação e teste do trabalho (verificar com o estagiário PAE, caso não saiba escrever um Makefile). Se necessário, incluam no .zip um arquivo chamado readme com informações que julgarem necessárias.

Os códigos deverão ser compilados pelo compilador gcc com a flag -std=c99. A não conformidade das implementações com a versão C estabelecida acarretará em nota zero. Atenção! Todos os códigos

enviados passarão pelo sistema de verificação de plágio. Se forem identificados códigos duplicados, todos os alunos envolvidos receberão nota zero.

5. Correção e Avaliação

As implementações serão avaliadas por meio de casos de testes, com peso 7, e pela legibilidade e boas práticas de programação, com peso 3.
Os seguintes casos implicarão em nota zero:

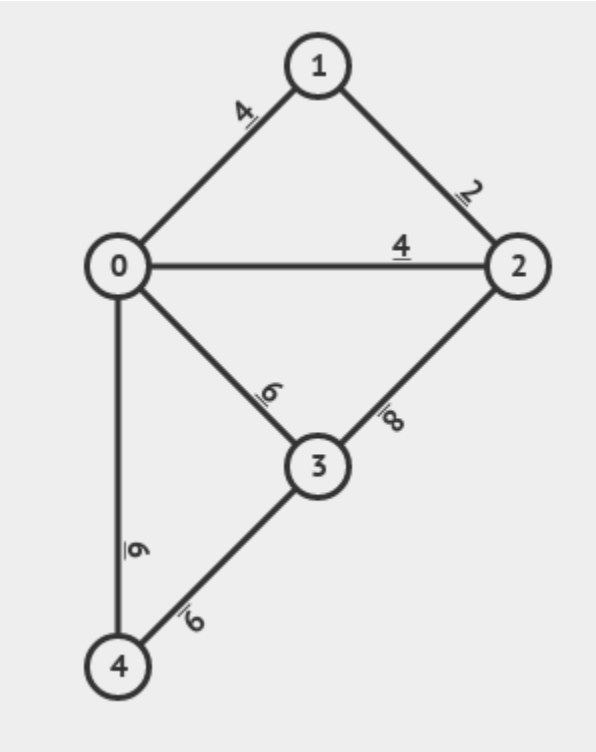
- Não conformidade com a versão C99;
- Programas não estruturados como um TAD;
- Exercícios plagiados.

Apêndice: Exemplos

Os comentários são apenas descritivos. Estes não existirão nas entradas e nem deverão ser impressos como saída.

Dado o digrafo:

```
5 7
0 1 4
0 2 4
0 3 6
0 4 6
1 2 2
2 3 8
3 4 9
```



Deste modo, o programa terá como saída as arestas em ordem de inclusão na árvore

```
(0,1)(1,2)(0,3)(0,4)
```