

Helton Hideraldo Bísaro

Algoritmos e Estruturas de Dados II -

Exercício Programa 1

2019

-

Apresentação:

O objetivo deste exercício-programa é implementar o algoritmo de Prim para árvores geradoras mínimas (AGM) em grafos. Basicamente, o algoritmo recebe um grafo $G = (V, E)$ e devolve uma árvore $T = (S, E')$. O programa deverá ser desenvolvido em C (compilável no gcc/mingw) e deverá ser executável via linha de comando. A chamada do programa será na forma:

prim.exe <arquivografo> <arquivosaida> Exemplo:

Prin.exe grafo01.txt saida01.txt

Implementação:

Para a implementação do algoritmo de Prim, recomenda-se usar como base o pseudo-código abaixo. Inicialmente S e E' estão vazios. Adicionamos um vértice inicial v e vamos acrescentando vértices em T na ordem mais barata possível. A função $p(v, u)$ representa o peso da aresta (v, u) .

```
Data:  $G = (V, E)$ 
1  $E' = \emptyset$  ;
2  $S = \emptyset$  ;
3 foreach vertex  $v \in V$  do
4    $\text{custo}[v] = \infty$ ;
5    $\text{ant}[v] = -1$  ;
6  $\text{custo}[v_0] = 0$  ;
7 while  $S \neq V$  do
8   Selecione  $u$  em  $V - S$  tal que  $\text{custo}[u]$  é mínimo;
9   Adicione  $u$  em  $S$ ;
10  foreach vizinho  $w$  de  $u$  do
11    if  $\text{custo}[w] > p(u, w)$  then
12       $\text{custo}[w] = p(u, w)$ ;
13       $\text{ant}[w] = u$ ;
```

Entrega do trabalho:

O trabalho deverá ser individual e ser entregue um único módulo em C, nomeado na forma:

d-<numerousp1>.c onde numerousp1 correspondem ao número USP do autor(a).

Exemplo: d-1234567.c

Este módulo conterá todas as funções implementadas pela dupla. Incluir nas linhas iniciais do programa o nome e número USP do autor(a).

Formatos dos arquivos:

<arquivografo>: Conterá a representação do grafo, e terá o seguinte formato: a 1ª linha conterá o número de vértices e arestas; as demais linhas conterão as especificações das arestas. Mais especificamente, o formato do arquivo será: n m o_1 d_1 $custo_1$ o_2 d_2 $custo_2$... onde: n , m (inteiros): número de vértices e de arestas, respectivamente. o_i d_i (inteiros): origem e destino da i -ésima aresta ($i = 1...m$) $custo_i$ (float ou double): custo da i -ésima aresta ($i = 1...m$) Observações:

1. Os vértices do grafo serão indexados de 0 a $n-1$. 2. Todos os campos são separados por espaços.

<arquivosaida>: Conterá os resultados da execução do programa. A 1ª linha conterá o custo da árvore geradora mínima (ou seja, a soma dos custos das arestas da árvore). As linhas seguintes conterão as arestas da árvore geradora mínima.

O código-fonte deverá ser compilável no gcc ou no mingw.

O trabalho deverá ser postado no tidia. O prazo para entrega é 30/04/2019.

Além da correção do programa, será considerada a qualidade da documentação do código-fonte.

Evidência de plágio entre trabalhos não apenas implicará na nota zero no trabalho, como também sujeitará os alunos envolvidos às medidas disciplinares cabíveis.

Observações finais:

Nos casos em que a AGM não é única, as arestas selecionadas por cada implementação poderão ser eventualmente diferentes.