

Resolução – Lista 6 (Projeto e Análise de Algoritmos)

Março - 2023 / Leticia Bossatto Marchezi – 791003

Questão 1

No algoritmo de Kruskal, explique o que são e como funcionam as primitivas `Make_Set(v)`, `Find_Set(v)` e `Union(u, v)` descrevendo o pseudo-código de cada uma delas.

Resolução:

As primitivas `Make_Set(v)`, `Find_Set(v)` e `Union(u, v)` constituem parte essencial no algoritmo de Kruskal para o problema da árvore geradora mínima. Suas funcionalidades são: - `Make_Set(v)`: Instancia uma nova árvore vazia, sendo seu pai o próprio nó e de altura 0. - `Find_Set(v)`: Retorna recursivamente à raiz da árvore, verificando se o nó atual é o próprio pai dele, condição definida na primitiva `Make_Set(v)`. - `Union(u,v)`: Une as raízes das árvores `u` e `v`. A árvore com maior rank se torna pai da outra árvore.

Questão 2

Execute o algoritmo de Kruskal no grafo a seguir, mostrando o trace completo (passo a passo). Qual é a MST obtida? Quanto vale seu peso?

Resolução:

$$T(n) = 2T(n - 1) + O(1) \quad (2.1)$$

Questão 3

Mostre que o algoritmo de Kruskal é ótimo, ou seja, sempre retorna uma MST de G .

Resolução:

Questão 4

. Mostre que a complexidade do algoritmo de Kruskal é $O(m \log n)$.

Resolução:

Questão 5

Explique o funcionamento do algoritmo de Prim. Porque ele é considerado um algoritmo guloso?

Resolução:

Questão 6

Execute o algoritmo de Prim no grafo a seguir com raiz no vértice I, mostrando o trace completo (passo a passo). Qual é a MST obtida? Quanto vale seu peso?

Resolução:

Questão 7

Demonstre a propriedade do corte, que relaciona um subconjunto S de V , a aresta de menor peso com uma extremidade em S e uma MST de G .

Resolução:

Questão 8

Prove a otimalidade (corretude) do algoritmo de Prim.

Resolução: