Universidade Federal de Pernambuco
Centro de Informática
Bach. em Engenharia da Computação

IF672ec Algoritmos e Estr. de Dados 2018.1
Prof. Paulo Fonseca

Prova Final — 12 de Julho de 2018

QUESTÃO 1

Considere o algoritmo a seguir.

Algorithm *hts*
Input $V = (v_0, \ldots, v_{n-1})$ vetor de n inteiros decimais de m dígitos cada.
Output ? ? ?

```
for d ← 0, ..., m − 1 do
    T ← ht_init (d)
    for i ← 0, ..., n − 1 do
        ht_insert (T, V[i])
    i ← 0
    for j ← 0, ..., 9 do
        cur ← T[j]
        while cur ≠ ⊥ do
            V[i] ← cur → val
            cur ← cur → next
            i ← i + 1
```

Sabendo que

- A chamada à função $ht\_init(d)$ retorna uma hash table vazia de tamanho 10 e com função de hash $h(k) = \lfloor k/10^d \rfloor \mod 10$. Essa tabela emprega a política de hashing aberto com listas encadeadas (sem sentinela).
- A chamada à função $ht\_insert(T, k)$ insere a chave k na hash table T. A inserção é sempre feita ao final da lista correspondente e em tempo constante.

a) Ilustre a execução do algoritmo sobre a entrada

$$V = (435, 987, 960, 347, 745, 664, 091, 514, 020, 260),$$

exibindo T e V ao final de cada iteração do laço mais externo.

b) *O que* o algoritmo faz? (máx 01 linha)

c) Qual o custo assintótico exato do algoritmo no pior caso? Justifique. (máx 05 linhas)

Solução

QUESTÃO 2

a) Desenhe a Árvore AVL cuja enumeração dos nós em *pós-ordem* é

10, 30, 20, 50, 70, 60, 90, 80, 40.

b) Represente a inserção do valor 65 na árvore do item (a), ilustrando a árvore logo após a inserção (antes das rotações) bem como após cada rotação que se fizer necessária.

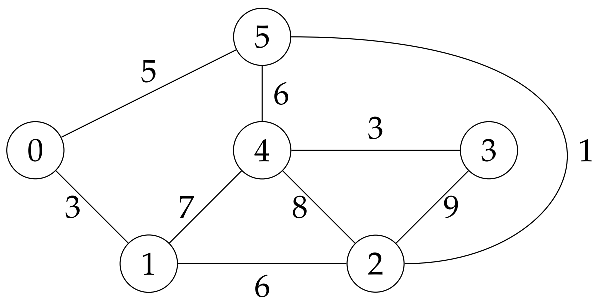
Solução

QUESTÃO 3 (2,0pt)

O Algoritmo de Kruskal é um algoritmo guloso para encontrar uma árvore geradora de custo mínimo (MST) de um grafo ponderado que faz uso da estrutura de dados *union-find* da seguinte maneira:

1. Inicie com a MST vazia e com cada vértice u numa componente isolada da union-find, `make_set(`u`);`
2. Ordene as arestas por 1º. peso, 2º. menor vértice, e 3º. maior vértice;
3. Para cada aresta (u, v) com peso w na ordem acima, se u e v não estão na mesma componente, una-as `union(`u`,`v`)`, e adicione a aresta à MST.

Considerando a execução do Algoritmo Kruskal sobre o grafo



represente graficamente a estrutura *union-find* C ao final da execução do algoritmo, considerando o emprego das heurísticas de *união ponderada* e *compressão de caminhos*.

Solução

■ **QUESTÃO 4** (2,0pt)

Considere o problema de encontrar o comprimento L da maior subsequência crescente (m.s.c.) de uma sequência de n inteiros distintos $S = \langle s_1, s_2, \ldots, s_n \rangle$, isto é, o maior L tal que existem índices $1 < i_1 < i_2 < \cdots < i_L < n$ com $s_{i_1} < s_{i_2} < \cdots < s_{i_L}$.

O seguinte algoritmo baseado na técnica de ____________________(a) resolve esse problema.

Seja L^k o comprimento da m.s.c. dentre os primeiros k elementos de S, $s_1, \ldots, s_k$. Queremos descobrir $L = L^{\_\_\_\_\_\_(b)}$. Para $k = 0$, Temos $L^0 =$__________(c). Para $k > 0$, temos dois casos. (I) Caso a m.s.c. dentre os k primeiros elementos *não* termine com o s_k, temos $L^k =$__________(d). (II) Caso não termine, então, definindo $pred(i) = \max j < i$ tal que $s_j < s_i$, se tal j existir, ou $pred(i) = 0$, do contrário, temos $L^k =$__________(e). Reunindo os casos (I) e (II), temos $L^k =$____________________(f). O algoritmo computa então os valores de L^k para $k = 0, \ldots$ e armazena os resultados numa tabela. Ilustre a execução do algoritmo sobre a entrada $S = \langle 2, 1, 5, 7, 4, 9, 3 \rangle$ exibindo a tabela computada e indicando claramente o resultado.

Solução

■ QUESTÃO 5 (2,0pt)

Um *super-sorvedouro* num grafo dirigido $G = (V, E)$ com n vértices numerados $0, \ldots, n-1$, e sem autoarestas (u, u), é um vértice com grau de saída 0 e grau de entrada $n-1$. Escreva em pseudocódigo um algoritmo $O(n)$ no pior caso que recebe G codificado como uma matriz de adjacências, e retorna o super-sorvedouro, se existir, ou então retorna -1 (só pode existir no máximo um).