

“A compreensão das questões faz parte da avaliação”

Análise e Projeto de Algoritmos

Bacharelado em Ciência da Computação

2ª Avaliação Individual

8 de junho de 2015

Nome: _____ Período: _____

Questão 1: (1.5 ponto)

Dado um polinômio $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$, com $a_n, a_{n-1}, \dots, a_1, a_0$ e $x \in \mathbb{R}$, utilize indução para desenvolver um algoritmo que calcula o valor de $P(x)$ aplicando exatamente n multiplicações e n adições.

Questão 2: (1.5 pontos)

Utilize programação dinâmica para desenvolver um algoritmo eficiente que determina elementos da sequência de Fibonacci $(0, 1, 1, 2, 3, 5, 8, 13, \dots)$.

Questão 3: (2.0 pontos)

O problema da mochila fracionária consiste em: dados vetores naturais (p_1, p_2, \dots, p_n) , (v_1, v_2, \dots, v_n) e um número natural c , encontrar um vetor racional (x_1, x_2, \dots, x_n) que maximize $x \cdot v$ (produto escalar x por v) sob as restrições $x \cdot v \leq c$ e $0 \leq x_i \leq 1$ para todo i . Escreva um algoritmo guloso eficiente que resolve este problema.

Questão 4: (2.0 pontos)

Sabemos que `Constroi-Heap()`, definido a seguir tem complexidade de pior caso $O(n \lg n)$. Mas esta estimativa é excessivamente folgada. Então, utilize análise amortizada para mostrar que a complexidade é apenas $O(n)$.

`Constroi-Heap(A)`

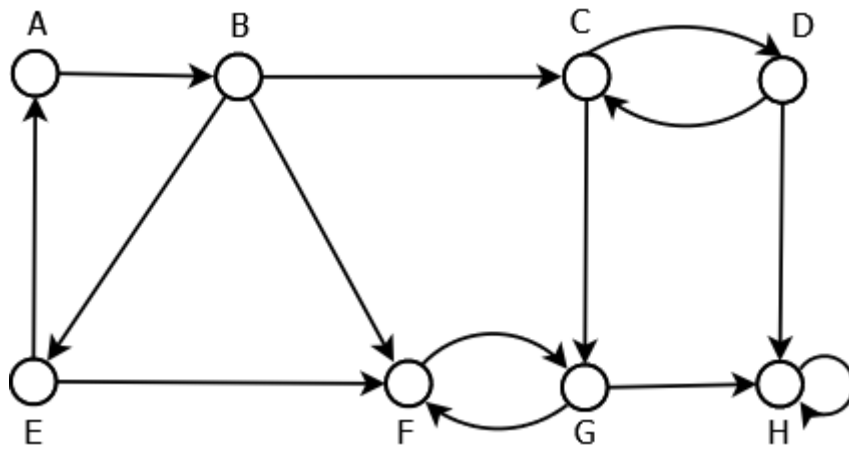
1. para i de $\text{chão}(n/2)$ até 1 faça
2. `Desce-Heap(A, n, i)`

`Desce-Heap(A, n, i)`

1. enquanto $(2i \leq n)$ faça
2. $f \leftarrow 2i$
3. se $(f < n)$ e $(A[f+1] > A[f])$
4. $f \leftarrow f+1$
5. se $(A[i] < A[f])$
6. $A[i] \leftrightarrow A[f]$
7. $i \leftarrow f$
8. senão
9. $i \leftarrow n+1$

Questão 5: (2.0 pontos)

Mostre (passo a passo) como a busca em profundidade ocorre no grafo a seguir. Suponha que o procedimento DFS considera os vértices em ordem alfabética e que a lista de adjacência também está em ordem alfabética. Não esqueça de indicar os tempos de descoberta e término de cada vértice do grafo.



DFS(G)

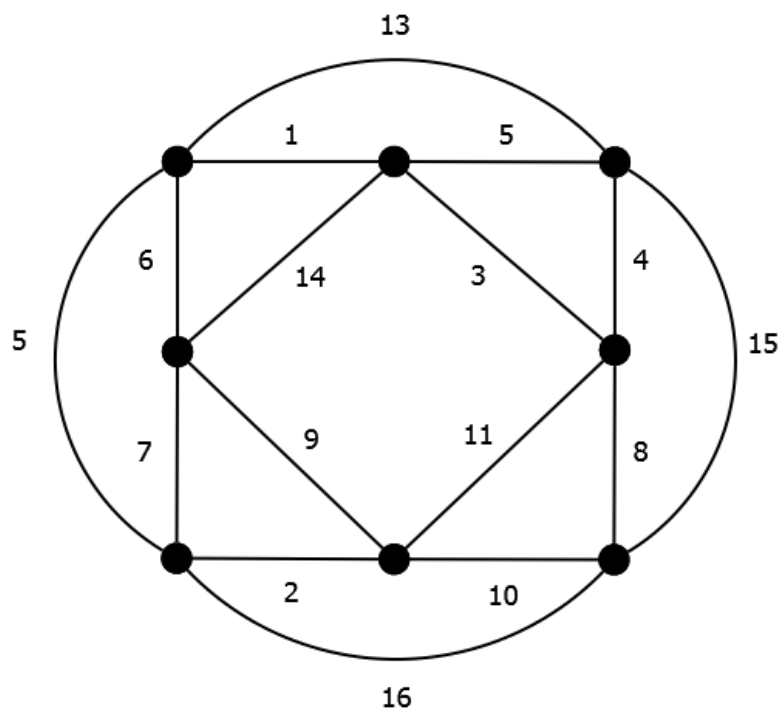
1. para cada vértice u em V[G] faça
2. cor(u) <- branco
3. tempo <- 0
4. para cada vértice u em V[G] faça
5. se cor(u) = branco
6. DFS-Visit(u)

DFS-Visit(u)

1. cor(u) <- cinza
2. tempo <- tempo+1
3. d(u) <- tempo
4. para cada vértice v em Adj(u) faça
5. se cor(v) = branco
6. DFS-Visit(v)
7. cor(u) <- preto
8. tempo <- tempo+1
9. f(u) <- tempo

Questão 6: (1.0 ponto)

Determine (mostrando passo a passo) a árvore geradora mínima do grafo a seguir, através da aplicação do algoritmo de Prim.



```
MST-Prim(G, w, r)
1. para cada vértice u em V[G] faça
2.     chave(u) <- infinito
3.     P(u) <- null
4. chave(r) <- 0
5. Q <- V[G]
6. enquanto Q != vazio faça
7.     u <- extrai-mínimo(Q)
8.     para cada vértice v em Adj(u) faça
9.         se (v pertence a Q) e (w(u,v) < chave(v))
10.            P(v) <- u
11.            chave(v) <- w(u,v)
```

“Esta avaliação terá duração máxima de 3 horas”