Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Ciência da Computação

IF672cc Algoritmos e Estruturas de Dados — 2013.1
Profs. Katia Guimarães, Paulo Fonseca

SEGUNDA CHAMADA — 02 de SETEMBRO de 2013

Text

- Esta prova contém 04 (quatro) questões dividida em duas partes.
- Resolva cada parte em uma folha separada.
- A duração da prova é de 2h00min.

PARTE I

QUESTÃO 1 (2,5pt)
Escreva em pseudo-código um algoritmo que recebe como entrada a raiz de uma árvore de busca binária T e dois valores $a \leq b$ e imprime todos os elementos de T com valor x tal que $a \leq x \leq b$, *em ordem crescente*. Justifique brevemente (máx 10 linhas) a complexidade do seu algoritmo e dê um exemplo do pior caso.

QUESTÃO 2 (2,5pt)
A operação max_heap_delete(H, i) remove o elemento da posição i da max heap H (representada como array). Escreva, em pseudo-código, o procedimento max_heap_delete com tempo de execução $O(\lg n)$, onde n denota o número de elementos de H.

Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Ciência da Computação

IF672cc Algoritmos e Estruturas de Dados — 2013.1
Profs. Katia Guimarães, Paulo Fonseca

SEGUNDA CHAMADA — 02 de SETEMBRO de 2013

- Esta prova contém 04 (quatro) questões dividida em duas partes.
- Resolva cada parte em uma folha separada.
- A duração da prova é de 2h00min.

PARTE II

QUESTÃO 3 (2,5pt)

a) Dê exemplo de um grafo com 5 vértices e 7 arestas no qual
   - 5 arestas têm peso positivo e 2 arestas têm pesos negativos e
   - Não há aresta incidente ao vértice 1 com peso negativo, e
   - O Algoritmo Dijkstra (1) dá a resposta incorreta.

b) Diga em no máximo três linhas o que exatamente faz com que o Algoritmo Dijkstra faça a escolha da resposta errada.

QUESTÃO 4 (2,5pt)
Escreva um algoritmo para resolver o problema 0-1 Knapsack (Mochila) em tempo $\Theta(n \times W)$, onde n é a quantidade de itens da entrada e W é a capacidade da mochila. Justifique detalhadamente o custo do seu algoritmo. O algoritmo deve ser escrito em JAVA ou no pseudo-código usado em sala de aula.

OBS: O problema 0-1 Knapsack define os itens que entrarão (1) ou não entrarão (0) na solução escolhida. (Ou seja, não há repetição ou fração de itens.)