

Filas de prioridades e Heapsort

5 de Dezembro de 2013

Exercício 1

Escreva a sequência de 13 chaves no vetor resultante da inserção da sequência de 3 chaves

46 77 45

na seguinte max-heap binária de tamanho 10:

99 88 59 52 60 23 10 36 38 58

Exercício 2

Escreva a sequência de 7 chaves no vetor resultante após realizar 3 operações sucessivas `delete-the-max` na seguinte max-heap binária de tamanho 10:

88 82 83 64 67 27 48 17 63 45

Exercício 3

Quais das afirmações a seguir sobre filas de prioridade são verdadeiras? Justifique. Assuma que a implementação é orientada ao máximo e usa vetores iniciando na posição 1.

- Uma heap ternária é uma representação de vetor de uma árvore completa com nós de aridade 3, onde a chave em cada nó é maior que ou igual às chaves de seus três filhos. Os filhos de $v[k]$ são $v[3k-1]$, $v[3k]$ e $v[3k+1]$.
- No melhor caso, deletar uma chave máxima de uma heap binária contendo $N \geq 100$ chaves consome apenas 2 comparações.
- Um programador pode preferir heapsort ao mergesort porque ele usa somente espaço constante além do array de entrada
- No heapsort, a principal razão para usar a construção da heap na forma bottom-up (usando demosções) é fazer menos comparações
- O número esperado de comparações para ordenar com o heapsort um vetor aleatório uniforme de N chaves distintas é aproximadamente $N \log N$