

Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Computação
Análise de Algoritmos
Listas de Prioridades

1. Seja S o *heap* com oito chaves especificado abaixo:

$$S = [92 \ 85 \ 89 \ 47 \ 52 \ 34 \ 20 \ 40].$$

Agora, considere as seguintes operações sobre S :

- Inserção das chaves 91 e 88 (nesta ordem); e
- Remoção da chave 92.

Escolha a opção abaixo que representa S após essas operações.

- (A) [91 85 89 47 52 34 20 40 88].
- (B) [89 91 34 85 52 20 40 47 88].
- (C) [91 88 89 85 52 34 20 40 47].
- (D) [89 47 52 34 85 88 91 20 40].
- (E) [91 88 85 40 47 52 89 34 20].

2. Uma solução para o problema da construção de um *heap* provém da observação de que todo vetor ordenado corresponde a um *heap*. Assim, um *heap* pode ser construído através da ordenação do vetor. Essa solução requer um número de passos $O(n \log n)$. Um método mais eficiente é descrito abaixo. Ele recebe um vetor $A[1..n]$ e acerta as prioridades dos nós internos da árvore em relação a seus descendentes a partir dos níveis mais baixos até a raiz. Seu tempo de execução pode ser limitado como $O(n)$.

MAX-HEAP (A, n)

1. para $i = n/2$ até 1 faça
2. HEAP-DECREASE-KEY (A, n, i)
3. fim

Seja um vetor dado pelas prioridades a seguir:

18 25 41 34 14 10 52 50 48

Encontre o *heap* obtido pela aplicação do algoritmo MAX-HEAP.

3. Considere o seguinte vetor S de números inteiros:

$$S = [10 \ 15 \ 11 \ 8 \ 3 \ 13 \ 12 \ 6 \ 1 \ 9 \ 2 \ 7 \ 5 \ 4].$$

Na execução do algoritmo MAX-HEAP, que transforma o vetor S em um *heap*, indique o número total de trocas realizadas.

- (A) 2
- (B) 1
- (C) 4
- (D) 3
- (E) 6

4. [6-1] Podemos construir um *heap* chamando repetidamente o procedimento HEAP-INSERT para inserir os elementos no *heap*. Considere a seguinte variação do algoritmo MAX-HEAP:

```
MAX-HEAP-ALT (A, n)
1. B[1] = A[1]
2. j = 1
3. para i = 2 até n faça
4.   HEAP-INSERT (B, j, A[i])
5.   j = j + 1
6. retorne (B)
```

a) Os algoritmos MAX-HEAP e MAX-HEAP-ALT sempre criam o mesmo *heap* quando são executados sobre o mesmo vetor de entrada? Prove que isso ocorre ou, então, dê um contra-exemplo.

b) Mostre que, no pior caso, MAX-HEAP-ALT requer o tempo $\Theta(n \log n)$ para construir um *heap* de n elementos.

5. Verifique se estas sequências correspondem ou não a um *heap*:

33 32 28 31 26 29 25 30 27

33 32 28 31 29 26 25 30 27

Caso a sequência não seja um *heap*, determine o *heap* obtido pela aplicação de um algoritmo de construção.

6. [POSCOMP 2003] Em um *heap* com n vértices existem

- (A) exatamente $\lfloor n/5 \rfloor$ folhas.
- (B) aproximadamente $\log(n)$ folhas.
- (C) não mais que $\lfloor n/5 \rfloor$ folhas.
- (D) exatamente $\lceil n/2 \rceil$ folhas.
- (E) não menos que $2n/3$ folhas.

7. Seja S o *heap* especificado a seguir:

$S = [92 \ 91 \ 90 \ 47 \ 85 \ 34 \ 20 \ 40 \ 46]$.

Determine o *heap* resultante da alteração de prioridade do seu primeiro nó de 92 para 53.