

Universidade Federal do Maranhão
Departamento de Informática
Curso de Ciência da Computação
I Avaliação - Estrutura de Dados II (2014.2)

Aluno:

1. Com relação ao método de ordenação Quicksort
 - a. Mostre passo a passo a ordenação sobre o conjunto $A = \{8, 9, 3, 4, 1, 2, 5, 7, 6\}$
 - b. Apresente um pseudocódigo recursivo usando um procedimento auxiliar `<int> Particione (<void> *vetor, <int> limiteBaixo, <int> limiteAlto)` que recebe um vetor de elementos de qualquer tipo e realiza o *pivoteamento* de um elemento entre os *limiteBaixo* e o *limiteAlto* (inicialmente 1 e N, respectivamente), retornando o índice do vetor onde o pivô foi posicionado. Essa posição deve ser usada como referência para as próximas chamadas recursivas de `<void> quickSort (<void> * vetor, <int> limiteBaixo, <int> limiteAlto)`
 - c. Faça uma análise da complexidade do algoritmo Quicksort, admitindo hipoteticamente que o total de elementos a serem ordenados é $n = 2^m$
2. Seja um heap de N elementos armazenados em um vetor,

```
typedef struct {  
    INFO vetor [N]  
} HEAP;
```

Faça um procedimento `INFO Remove (HEAP h, INT i)` que remova e retorne o i-ésimo elemento ($i \leq N$) do heap, ajustando-o após a remoção. Faça ainda uma análise de eficiência desse procedimento, considerando o número de deslocamentos de elementos para ajustar o heap nos melhores e piores casos.

3. O que dizer de dois programadores que relatam terem desenvolvido algoritmos de ordenação, A1 e A2, respectivamente, ambos baseados em comparação, supostamente de complexidade:
 - a. $A1(n) = n^{2/3}$
 - b. $A2(n) = n^{3/2}$

Análise à luz do que existe de melhor em termos de complexidade de algoritmos de ordenação baseados em comparação.

BOA PROVA