

U.C. 21046

Estruturas de dados e algoritmos fundamentais

09 de Julho de 2014

INSTRUÇÕES

Para a resolução deste **p – Fólio** aconselha-se que:

- Verifique o exemplar que lhe foi entregue e, no caso de estar incompleto ou com qualquer deficiência, dirija-se ao professor vigilante.
- O **p-fólio** é composto por 3 questões.
- O teste termina com a palavra **FIM**.
- Utilize, sempre, uma letra legível e não use uma caneta de outra cor que não seja o preto ou o azul - as respostas a lápis não serão consideradas.
- Tenha em atenção que o **p-Fólio** tem a duração de 1 hora e 30 minutos.

Critérios de avaliação e cotação

- Deve assinalar todas as opções tomadas: no código dos seus programas, todas as constantes, variáveis, métodos ou funções devem ser devidamente explicadas através de comentário
- As respostas, que embora, sintáctica e semanticamente corretas, se apresentem pouco estruturadas serão severamente penalizadas, ou não consideradas.
- As respostas sem justificação serão fortemente penalizadas
- As respostas de conteúdo inadequado não serão consideradas.

1º Questão (4 Valores)

a) Qual a complexidade dos seguintes ciclos? Justifique

a1)

```
for (i=0; i<n; i++)  
    for (j=0; j<n; j++)  
        a[i][j]=b[i][j]+c[i][j];
```

$O(n^2)$

a2)

```
for (cnt2=0, i=1; i<=n; i++)  
    for (j=1; j<=i; j++)  
        cnt2++;
```

$O(n^2)$

b) Assuma que lhe é dada uma tabela que contém N números. Pretende-se determinar o comprimento da maior sub-sequência crescente existente na tabela. (Se a tabela contiver os números: 2,4,7,6,5,8,10,11,4 então o comprimento da maior sub-sequência crescente é 4 que corresponde à sub-sequência: 5,8,10,11). Elabore um algoritmo eficiente para a resolução do problema e indique a sua complexidade.

```
int MaiorSubSeq(int * vec, int N)  
{  
    int maior=1;  
    int atual=1;  
    for (int i=1; i<N; i++) // começa na posição 1 e não na posição 0  
        if (vec[i]>vec[i-1]) // sequencia crescente  
            atual++;  
        else  
        {  
            if (maior<atual)  
                maior=atual;  
            atual=1;  
        }  
    return maior;  
}
```

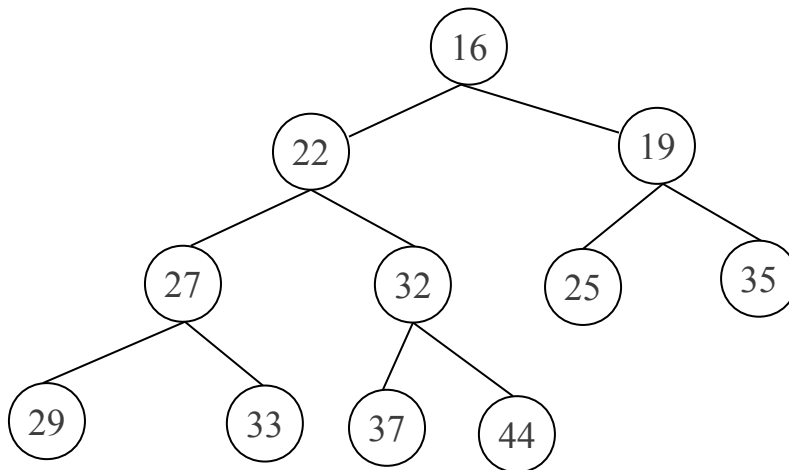
A complexidade é $O(n)$ pois o vetor é percorrido uma vez em todas as posições

2º Questão (4 Valores)

Considere o seguinte Heap no qual a prioridade é maior quanto menor o valor.

16	22	19	27	32	25	35	29	33	37	44
----	----	----	----	----	----	----	----	----	----	----

a) Desenhe a árvore binária correspondente a este Heap e explique porque motivo a condição da prioridade é mantida.



A condição de prioridade é mantida pois a prioridade de cada nó ascendente é sempre maior do que a dos descendentes (cada nó ascendente tem um valor mais baixo do que os descendentes)

b) Uma árvore binária diz-se regular se todo o nó que não é folha possui exatamente 2 filhos. Implemente a uma função função **Regular** que indica se um Heap implementado em vetor é regular.

Num Heap implementado em vetor observam-se as seguintes regras:

Filho esquerdo do nó i está na posição $2i$

Filho direito do nó i está na posição $2i+1$

Pai do nó j está na posição $j/2$

```

bool function Regular(int heap[N])
{
  for(int i=1; i<=N;i++)
    if ((2*i<=N) && (2*i+1>N)) // existe um nó com filho esquerdo e sem filho direito
      return false;
  return true; // todos os nós possuem 2 filhos ou então são folhas
}
  
```

3º Questão (4 Valores)

a) Considere o seguinte vetor de inteiros: 6 3 2 9 8 1. Represente os passos seguidos para ordenação do vetor utilizando o algoritmo Selection Sort (ordenação por seleção).

6 3 2 9 8 1

1 6 3 2 9 8 (ao fim da 1º passagem)

1 2 6 3 9 8 (ao fim da 2º passagem)

1 2 3 6 9 8 (ao fim da 3º passagem)

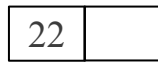
1 2 3 6 9 8 (ao fim da 4º passagem)

1 2 3 6 8 9 (ao fim da 5º passagem)

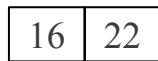
b) Considere uma árvore B+ de ordem 3 e capacidade 2. Represente esquematicamente a evolução da árvore após cada uma das seguintes operações de inserção: I 22 ; I 16; I 41; I 88; I 8; I 22

A raiz desta árvore ou é folha ou possui entre 2 e 3 filhos
O número de chaves varia entre 1 e 2
O número de elementos de cada folha varia entre 1 e 2

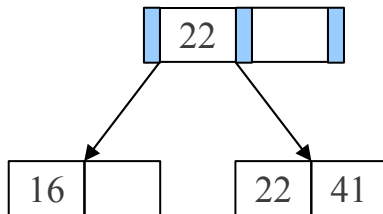
Após I 22



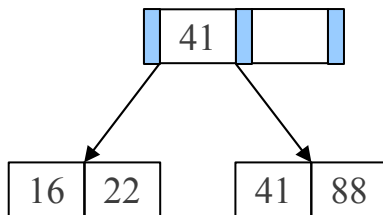
Após I 16



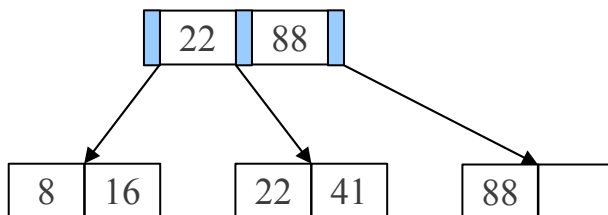
Após I 41



Após I 88



Após I 8



FIM