

SIN5013 - Exemplo de Prova do Conteúdo Semestral

1. Encontre as equações de recorrência dos seguintes métodos (não é necessário resolver, mas é preciso identificar precisamente suas equações de recorrência [desejamos identificar **exatamente** quantas vezes os **comandos hachurados** serão executados, **no pior caso**, em relação a n]). Não é necessário resolver as equações, apenas identifica-las:

```
static int busca (int[] A, int x, int ini, int fim){  
    if (ini>fim) return -1;  
    int meio = (fim+ini)/2;  
    if (A[meio]==x) return meio;  
    int i1 = busca(A, x, ini, meio-1);  
    int i2 = busca(A, x, meio+1, fim);  
    if (i1>i2) return i1;  
    return i2;  
}
```

(neste caso $n = \text{fim} - \text{ini} + 1$)
Equação de Recorrência:

```
void hanoi2(char ori, char dst, char aux, int n) {  
    if(n <= 3) {  
        System.out.print("Move de " + ori + " para " + dst);  
    }else {  
        hanoi2(ori, aux, dst, n/3);  
        hanoi2(ori, dst, aux, n/3);  
        hanoi2(aux, dst, ori, n/3);  
    }  
}
```

Equação de Recorrência:

2. Resolva a equação de recorrência a seguir e identifique sua complexidade assintótica θ (não é preciso provar/demonstrar a complexidade assintótica):

$$T(n) = 2 \cdot T(n/2) + 1, \quad T(1) = 1$$

3. Escreva um programa (utilizando um ou mais métodos/funções em código ou pseudo-código) que dado um grafo não direcionado com n nós **retorne o número de componentes conexos do grafo**. Este grafo pode ser representado por: (a) uma matriz de adjacências booleana com $n \times n$ células correspondendo aos n nós do grafo; ou representado por: (b) um arranjo de nós sendo que cada nó possui um arranjo de vizinhos; neste caso, considere que `Nó` é uma classe ou estrutura que possui os seguintes campos [ou atributos]: `numVizinhos` (campo do tipo inteiro com o número de vizinhos do respectivo nó), `vizinhos` (arranjo de ponteiros para Nós com `numVizinhos` elementos), `visitado` (campo booleano). Escolha a representação que te convém, se desejar, considere que todas as variáveis são globais/*static*. Assuma que as variáveis já estão devidamente inicializadas/preenchidas.

<i>Representação 1</i>	<i>Representação 2</i>
<pre>int numNos = n; boolean adjacencias[][] = new boolean[n][n]; boolean visitados[] = new boolean[n];</pre>	<pre>int numNos = n; No nos[] = new No[numNos]</pre>