

Lista de Exercícios 2 – ACH2002 – Introdução a Ciência da Computação II

1 (3,0). Dados os três métodos abaixo que calculam o valor máximo e mínimo de um arranjo:

```
// METODO ITERATIVO
public static int[] maximoMinimoIterativo(int[] A){
    int[] res = new int[2];
    res[0] = A[0]; // res[0] contem o maximo
    res[1] = A[0]; // res[1] contem o minimo
    for (int i=1;i<A.length;i++){
        if (A[i]>res[0]) res[0] = A[i];
        else {
            if (A[i]<res[1]){
                res[1] = A[i];
            }
        }
    }
    return res;
}

// METODO RECURSIVO (INDUÇÃO FRACA)
public static int[] maximoMinimoIndFraca(int[] A,int elementos){
    // Criterio de parada: só sobrou um número a ser avaliado
    if (elementos==1){
        int[] res = new int[2];
        res[0] = A[0]; // res[0] contem o maximo
        res[1] = A[0]; // res[1] contem o minimo
        return res;
    }else{
        elementos--;
        int[] res = maximoMinimoIndFraca(A,elementos);
        if (A[elementos]>res[0]){
            res[0] = A[elementos];
        }else{
            if (A[elementos]<res[1]){
                res[1] = A[elementos];
            }
        }
        return res;
    }
}

// METODO RECURSIVO (INDUÇÃO FORTE)
public static int[] maximoMinimoIndForte(int[] A,int ini, int fim){
    // Criterio de parada: soh sobrou um numero a ser avaliado
    if (ini==fim){
        int[] res = new int[2];
        res[0] = A[ini]; // res[0] contem o maximo
        res[1] = A[fim]; // res[1] contem o minimo
        return res;
    }else{
        int meio = (ini+fim)/2;
        int[] res = maximoMinimoIndForte(A,ini,meio);
        int[] res2 = maximoMinimoIndForte(A,meio+1,fim);
        if (res2[0]>res[0]) res[0] = res2[0];
        if (res2[1]<res[1]) res[1] = res2[1];
        return res;
    }
}
```

1a) Escreva a função $T(n)$ (número de vezes que as comparações que estão nas linhas destacadas dos “if”s são executadas para cada um dos métodos, em relação ao tamanho do arranjo de entrada), considerando sempre o pior caso de execução de cada método. Se o método for recursivo, escreva $T(n)$ como uma função de recorrência.

Respostas:

Método Iterativo: $T(n) =$

Método Recursivo (Indução Fraca): $T(n) =$

Método Recursivo (Indução Forte): $T(n) =$

1b) Calcule a complexidade assintótica (θ) para cada uma das funções $T(n)$ acima (no caso de $T(n)$ ser uma equação de recorrência, apresente o cálculo e não apenas o resultado).

2 (1,0). Utilize o teorema mestre para calcular a complexidade assintótica da seguinte equação de recorrência: $T(n) = 4 \cdot T(n/2) + n \cdot \log(n)$; $T(1) = 1$

Teorema Mestre

Sejam $a \geq 1$ e $b \geq 2$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida para os inteiros não-negativos pela relação de recorrência

$$T(n) = aT(n/b) + f(n)$$

Então $T(n)$ pode ser limitada assintoticamente da seguinte maneira:

- ❶ Se $f(n) \in O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) \in \Theta(n^{\log_b a})$
- ❷ Se $f(n) \in \Theta(n^{\log_b a})$, então $T(n) \in \Theta(n^{\log_b a} \log n)$
- ❸ Se $f(n) \in \Omega(n^{\log_b a + \epsilon})$, para alguma constante $\epsilon > 0$ e se $af(n/b) \leq cf(n)$, para alguma constante $c < 1$ e para n suficientemente grande, então $T(n) \in \Theta(f(n))$

3 (1,5). Considere o Problema da Mochila, definido a seguir. Dados: i) uma mochila que admite um certo peso; e ii) um conjunto de objetos, cada um com um valor e um peso; Selecione o conjunto de objetos que caibam dentro da mochila de forma a maximizar o valor total dentro da mochila.

Considere que você pode quebrar os objetos em frações e colocar apenas uma fração do objeto na mochila. Considere um algoritmo guloso para resolver este problema. A cada iteração este algoritmo colocará na mochila o objeto (ou uma fração do objeto) que pareça ser mais lucrativo para o algoritmo. Desta forma, o problema se resume a ordenar os objetos pelo quão lucrativo cada objeto é. Ordene os objetos a seguir em **ordem decrescente** em termos de lucratividade (do mais lucrativo para o menos lucrativo). A função de lucratividade será aquela que torne o algoritmo guloso ótimo para este problema.

Objeto:	O1	O2	O3	O4	O5	O6	O7
Peso:	5	14	9	80	8	16	24
Valor:	10	14	18	20	24	16	12

3a) Resposta (preencha a tabela abaixo com os objetos após a ordenação):

Objeto:							
----------------	--	--	--	--	--	--	--

3b) Qual deve ser a função de lucratividade usada?

4 (0,5). Demonstre que $n \in o(n^2)$

5 (1,5). Você deseja ordenar um conjunto de n números, cujos valores variam de 1 a n^2 . Qual a complexidade assintótica θ (no caso médio) dos seguintes algoritmos para se ordenar este conjunto? Considere que cada valor do conjunto a ser ordenado tem uma quantidade de no máximo $\log(n)$ dígitos.

BubbleSort:

CountingSort:

RadixSort:

6 (1,0). Qual dessas funções podem ser utilizadas como função hash (no sentido de ocuparem potencialmente todas as posições da tabela hash)? Considere que a chave x é um valor inteiro positivo, e uma tabela hash de 200 elementos.

Sem considerar endereçamento aberto:

[] $f(x) = 5 \% 201$;

[] $f(x) = x \% 201$;

[] $f(x) = x * 5 \% 201$;

[] $f(x) = x * 2 \% 201$;

[] $f(x) = x * 3 \% 201$;

[] $f(x) = x$;

Considerando endereçamento aberto:

[] $f(x,i) = (x+i) \% 201$

[] $f(x,i) = (x+201*i) \% 201$

[] $f(x,i) = (x+i) \% 201$

7 (0,5). Quais são os três métodos que uma estrutura hash deve ter (apenas cite-os)?

Resposta:

Método 1:

Método 2:

Método 3:

8 (1,5). Calcule a complexidade assintótica (θ) da função de recorrência $T(n)$ a seguir, NÃO UTILIZE O TEOREMA MESTRE. Exiba o cálculo e não apenas o resultado.

$T(1) = 10$

$T(n) = 3 * T(n/3) + n$

8. O que é um algoritmo de ordenação estável? Dê um exemplo.

9. O que é um algoritmo de ordenação *in place*? Dê um exemplo.

10. Dados os algoritmos de ordenação baseados em comparação vistos em aula. Qual deles tem uma menor complexidade em relação ao número de atribuições? E em relação ao número de comparações (considere sempre o pior caso).

11. O que é agrupamento primário (em estruturas hash)?

12. Qual a diferença entre endereçamento direto e hash? Cite uma vantagem e uma desvantagem da estrutura hash.