

Prova 1 – ACH2002 – Introdução a Ciência da Computação II

Nome _____ NUSP _____

1 (1,5). Dadas seguintes classes em Java (Pessoa e Aluno):

```
package prova1;
public class Pessoa {
    public static int contador = 0;
    public Pessoa() {
        contador++;
    }
    public void estudar() {
        System.out.println("Pessoa estudando.");
    }
    protected void comer() {
        System.out.println("Pessoa comendo.");
    }
    void caminhar() {
        System.out.println("Pessoa caminhando.");
    }
    private void correr() {
        System.out.println("Pessoa correndo.");
    }
}
```

```
package prova1;
public class Aluno extends Pessoa{
    public void estudar() {
        System.out.println("Aluno estudando.");
    }
    protected void comer() {
        System.out.println("Aluno comendo.");
    }
    private void correr() {
        System.out.println("Aluno correndo.");
    }
}
```

1a) Assinale com um X quais linhas do programa abaixo não funcionarão:

[]	package prova1;
[]	public class ExecutarPessoas {
[]	public static void main(String[] args) {
[]	Pessoa p1;
[]	Pessoa p2;
[]	Aluno a1;
[]	Aluno a2;
[]	p1 = new Pessoa();
[]	p2 = new Aluno();
[]	a1 = new Aluno();
[]	System.out.println("Contador Al: " + a1.contador);
[]	System.out.println("Contador Pessoa: " + Pessoa.contador);
[]	a2 = new Pessoa();
[]	p2.estudar();
[]	p2.comer();
[]	p2.caminhar();
[]	p2.correr();
[]	}
[]	}

1b) Após excluir as linhas que não funcionam, escreva abaixo o que será impresso como resultado da execução desse programa?

2 (1,5). Considere o método a seguir que calcula os valores máximo e mínimo de um arranjo:

```
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;
}
```

2a) Calcule a função $T(n)$ (número de vezes que as comparações que estão nas linhas dos "if"s são executadas, em relação ao tamanho do arranjo de entrada) para o caso em que o arranjo de entrada está ordenado de forma crescente (melhor caso).

Resposta: $T(n) =$

2b) Complete o método a seguir (escreva o código) que calcula o máximo e o mínimo de um arranjo sabendo-se que SEMPRE o arranjo de entrada está ordenado em ordem crescente (este código deve ter complexidade ótima para este problema específico).

```
public static int[] maximoMinimoNovo(int[] A){
    int[] res = new int[2];
```

```
        return res;
    }
```

3 (1,5). Utilize o teorema mestre para calcular a complexidade assintótica da seguinte equação de recorrência: $T(n) = T(n/2) + n \cdot \log(n)$; $T(1) = 1$

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 $T(n) \in \Theta(f(n))$

4 (1,0). Complete o método abaixo de maneira a implementar um método recursivo que, dado um inteiro $n > 1$, calcule e retorne o fatorial desse número.

```
public static int[] fatorial(int n){
```

5 (1,5) Calcule a complexidade assintótica (θ) da seguinte função de recorrência:
 $T(n) = T(n-1) + \log(n)$, $T(1) = 1$

6 (2,0). Assinale as alternativas CORRETAS:

- () Se $f(n) \in \theta(g(n))$ então $g(n) \in \theta(f(n))$
- () Se $f(n) \in O(g(n))$ então $g(n) \in \omega(f(n))$
- () $2^n \in O(n)$
- () $2^n \in O(n^n)$
- () $n^2 - n \in \omega(n)$
- () $n \in \omega(n)$
- () $n \in \omega(\log(n))$
- () $\log(n) + n^2 \in \Omega(\log(n))$
- () $n \cdot \log(n) \in O((\log(n))^2)$
- () $\log_{10} n \in \theta(\log_2 n)$

7 (1,0). Demonstre que $(n^2 - 2 \cdot n) \in \omega(n)$