DISCIPLINA	Análise de Algoritmo	**TURMA**	
NOME ALUNO		**Duração**	**90 minutos**
Nº DA MATRÍCULA		**DATA**	**04. 01. 2023**
NOME PROFESSOR	Judson Paiva		

Leia com atenção o enunciado e responda <u>SOMENTE</u> o que lhe foi pedido.

1 – [**3.0**] Considerando apenas a execução das operações relevantes, mostre a função de complexidade de tempo para cada procedimento apresentado (p1, p2 e p3):

```java
public void p1(int n){

    for (int i = 1; i < n; i = i * 2) {
        for (int j = 1; j < n; j = j * 2){
            System.out.println(i);
        }
    }
}
```

```java
public void p2(int n) {

    int i = 0;
    int j = 1;

    while (j < n) {
        j = j + i;
        i++;
    }
}
```

```java
public void p3(int n) {
    int m = 0;
    for (int i = 1; i < n; i = i * 2)
    {
        m++;
    }
    for (int j = 1; j < m; j = j * 2)
    {
        System.out.println(j);
    }

}
```

Resposta:

O($logn \cdot logn$) **O($\sqrt{n}$)** **O($loglogn$)**

2 – [**4.0**] Levando em contas as condições i, ii e iii, diga argumentado se é verdadeira ou falsa as alternativas apresentadas (a, b, c e d).

i. Todas as variáveis e constantes são inteiras e positivas, a menos que sejam explicitamente identificadas de outra forma;

ii. As funções $f(n)$ e $g(n)$ são positivas e $f(n) < g(n)$ do ponto de vista de crescimento assintótico;

iii. E $p(n) = \sum_{i=0}^{g} a_i n^i$ é um polinómio de grau g, as constantes a_i $(1 \leq i \leq g)$ reais, sendo $a_g \neq 0$ e k uma constante.

a) $p(n) = \Omega(n^k)$, se $k \leq g$

VERDADEIRA.
De acordo com a definição f(n) = Ω(g(n)) se existirem constantes positivas c e n0, tais que cg(n) ≤ f(n) para todo n ≥ n0. Tendo em conta isso, vamos supor que existem constantes positivas c e n0, tais que

cnk ≤agng +ag−1ng−1 +...+a0

e dividirmos a inequação por nk temos:

c≤agng−k +ag−1ng−k−1 +...+a0n−k.

Á medida que n cresce, cada termo do lado direito da inequação fica menor, mas positivo. Assim, é possível achar constantes c e n0 que satisfazem a inequação acima.

b) $p(n) = O(n^k)$, se $k \geq g$

VERDADEIRA.

O mesmo raciocínio de cima, mas agora quem cresce são os termos mais a esquerda. E por outra leve em conta a definição da notação **O**

c) Seja $\Omega(n^e) = X$) onde X representa o conjunto de funções que satisfaz a notação O para a função n^e. O conjunto $\{n^2, nlog\, n^2, n^{\pi/2} log\, n^2, 1/n^e\} \subset$ X. Onde e é a constante de Euler.

FALSA.

P conjunto X possui funções, que têm uma taxa de crescimento inferior que a ordem de n^e. Então X não pode ter como limite inferior n^e

d) Levando em conta todos os detalhes da derivada de $h(n) = n^2$ e de $l(n) = 4n^2 + 2n$. Podemos afirmar que $\forall n \geq 0$, $h(n)$ cresce mais lentamente que $l(n)$ e, portanto, $h(n) = O(l(n))$

Calcule a derivada de h(n) e também de l(n) e verifica se l(n) não cresce mais rápido. Portanto é **VERDADEIRA.**

3 – [**5.0**] Sejam $f(n)\ e\ g(n)$ funções assintoticamente positivas. Prove ou conteste para que valores de $n_0\ e\ c\ (ou\ c_1\ e\ c_2)$ cada uma das seguintes conjeturas.

a) [**1.0v**] $\binom{n}{3} = O(n^3)$

AFIRMAÇÃO VERDADEIRA. Desdobrando a fórmula teremos uma expressão do terceiro grau. Agora bastará encontrar os valores c e n zero que respectivamente podem ser propostos: 1 e 1/6

b) [**1.0v**] Se $f(n) = 1 + 2 + 3 + \cdots + n$ então $f(n) = n^2 + \Omega(n)$

VERDADEIRA.

Desdobra a função terão uma expressão do segundo grau, e tente estudar o comportamento assintótico com os valores para n=2 e c=1

c) [**1.5v**] $T(n) = T\left(\frac{n}{2}\right) + n + \theta(1)$, para n>1 e $T(1) = 1,$ mostre que T(n) = O(n)

VERDADEIRA.
Desdobra a função terão uma expressão do primeiro grau, e tente estudar o comportamento assintótico com os valores para n=1 e c=1

d) [**1.5v**] $T(n) = 2T\left(\frac{n}{2}\right) + O(n+1)$, para n>1 e $T(1) = 1,$ mostre que T(n) = $\theta(nlogn)$

VERDADEIRA.
Desdobra a função terão uma expressão igual a $nlogn$, e tente estudar o comportamento assintótico com os valores para n=1 e c=1

4 – [**3.0**] Observe o programa abaixo.

```
public void p4(int n){
    if(n>0){
        for(int i=1; i<n; i=i*2){
            System.out.println(i);
        }
        p4(n-1);
    }
}
```

b) **[1.0v]** Encontre a relação de recorrência para esse procedimento.

T(n) = T(n-1) + logn , n > 0 e T(1) = 1, n=0

c) **[3.0v]** Resolva a mesma recorrência

Solução: **O(logn!) ou então O(nlogn)**

Para mais informações sobre recorrência veja o link abaixo:
https://web.stanford.edu/class/archive/cs/cs161/cs161.1168/lecture3.pdf

5 – [**4.0**] Em Teoria dos números, a sequência de Padovan é o conjunto de inteiros P(n) definida pelos valores 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265

a) Implemente uma função recursiva que permita gerar o n-ésimo termo da sequência de Padovan

b) Encontre a relação de recorrência para esse procedimento recursivo que escreveste.

c) Estime a complexidade de tempo usando o método de substituição

Resposta:

Estudem a respeito disso toda base anterior resolvida lhe dá habilidades para resolver essa questão. Discutam em grupos (talvez de WhatsApp) a respeito.

Formulário:

$$c_{m,p} = \binom{m}{p} = \frac{m!}{p!(m-p)!}$$

$$s_n = \frac{(a_1 + a_n).n}{2}$$

$$S = \frac{b_1(q^n - 1)}{q - 1}$$