

1. (2 pontos) Prove por indução que $1 + 2 + 2^2 + 2^3 + \dots + 2^n = 2^{n+1} - 1$, $\forall n \geq 0$.

9.5

2. (2 pontos) Assinale as alternativas CORRETAS: (responda aqui mesmo)

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

3. (2 pontos) Um determinado algoritmo possui o tempo de execução descrito pela recorrência

$$8T(n/2) + 5n^3 + 10n^2$$

Qual a complexidade desse algoritmo? Calcule utilizando o Teorema mestre, enunciado abaixo.

PROVE em qual dos 3 itens do Teorema mestre a recorrência se encaixa.

Teorema Mestre

Sejam $a \geq 1$ e $b > 1$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida sobre os inteiros não negativos pela recorrência

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

onde interpretamos n/b com o significado de $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$

Então, $T(n)$ pode ser limitado assintoticamente como a seguir:

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$

2. Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \lg n)$

3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$,

e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para todo n suficientemente

4. (2 pontos) Considere a seguinte função recursiva (pseudocódigo)

```
f(x){  
    se (x = 1)  
        retorne -x  
    retorne -f(x/2) + x  
}
```

Qual é o retorno da chamada $f(16)$? (responda aqui mesmo): 9

5. (2 pontos) A Profa. Arianne possui uma lista de n EPs (Exercícios-programas) possíveis, com o tempo de estimado de implementação (em semanas) de cada um deles ($t[i]$, $1 \leq i \leq n$). Ela gostaria de propor o maior número possível de EPs na disciplina ACH2002, considerando um semestre de 15 semanas.

a) Apresente um programa (em pseudocódigo) o mais eficiente (em termos de tempo) possível para calcular esse número máximo de EPs e imprimir seus índices. Obs: assumo que você pode, antes do seu programa, ordenar ou não o vetor t como quiser (se o fizer, diga como está ordenado). O tempo para ordenação não deve ser considerado.

b) Qual a complexidade de seu algoritmo?