**APS 03 Daniela Araldi**

**1.**

**a)** 0+1+2+3+4+5 = 15

**b)** (2⁰)+(2¹)+(2²)+(2³)+(2⁴)+(2⁵) = 63

**c)** (0 + 2⁰) + (1 + 2¹) + (2 + 2²) + (3 + 2³) + (4 + 2⁴) + (5 + 2⁵) = 1 + 3 + 6 + 11 + 20 + 37 = 78

**d)** n + n + ... + n = n \* n = n²

**2.**

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **5n** | **4n²** | **2n-1** |
| 0 | 0 | 0 | 0,5 |
| 1 | 5 | 4 | 1 |
| **2** | **10** | **16** | 2 |
| 3 | 15 | 36 | 4 |
| 4 | 20 | 64 | 8 |
| 5 | 25 | 100 | 16 |
| 6 | 30 | 144 | 32 |
| 7 | 35 | 196 | 64 |
| 8 | 40 | 256 | 128 |
| 9 | 45 | 324 | 256 |
| **10** | 50 | **400** | **512** |
| 12 | 60 | 576 | 2048 |
| 14 | 70 | 784 | 8192 |
| 16 | 80 | 1024 | 32768 |
| 18 | 90 | 1296 | 131072 |
| 20 | 100 | 1600 | 524288 |

**\*Momento em que 4n² é CAS para 5n;**

**\*Momento em que 2n-1 é CAS para 4n².**

**3.**

**a)** 2n³ + n(n² - n - 1)

2n³ + n³ - n² - n

Complexidade O(n³)

**b)** (n² + 2)/n

n²/n + 2/n

n

Complexidade O(n)

**c)** n(n - 1)/2

(n² - n)/2

Complexidade O(n²)

**d)** n + n²log n

Complexidade O(n² log n)

**4.**

**a)** L1. Θn

L2. n Θ(n) = O(n²)

L3. n²n = O(n³)

L4. O(1) //constante porque é uma atribuição normal

O(n) + O(n²) + O(n³) = **O(n³)**

**b)** L1. O(n)

L2. n O(n) = O(n²)

L3. O(n²)

L4. n² O(n) = O(n³)

L5. O(n³)

O(n) + O(n²) + O(n²) + O(n³) + O(n³) = **O(n³)**

**c)** inteiro *max*(inteiro: vet[n])

L3. O(1)

L4. O(n)

L5. O(1)

L6. O(1)

Algoritmo XYZ

L3. O(n)

L4. O(1)

L5. O(1)

L7. O(1)

L8. n O(n)

L9. n O(n) = O(n²)

L10. n O(n²) = O(n³)

L11. O(1)

(7 \* O(1)) + (2 \* O(n)) + O(n²) + O(n³) = **O(n³)**

**5.**

Para duas funções g(n) e f(n) temos que f(n) = Θ(g(n)) ↔ f(n) = Ω(g(n)) e f(n) = O(g(n)).

f(n) = Θ(g(n)) é a cota assintótica exata : c.f(n) é CAS para g(n);

f(n) = Ω(g(n)) é a cota assintótica inferior : c.f(n) é CAS para g(n);

f(n) = O(g(n)) é a cota assintótica superior : c.g(n) é CAS para f(n).

**6.**

Sim, pois se o melhor caso é O(n log n), então ele representa o limite assintótico inferior (Ω(n log n)).

**7.**

**a)** Complexidade de tempo representa o tempo necessário para a execução de um algoritmo; complexidade de espaço representa a memória necessária para a execução de determinado algoritmo.

**b)** Complexidade pessimista trata-se de quando a solução do algoritmo se dá na última solução possível; complexidade média é a esperada, onde há um processamento médio para solução do algoritmo; quanto ao melhor caso, trata-se de quando o algoritmo encontra a solução na primeira iteração executada, gastando o mínimo de processamento.

**c)** Separa-se a função dominante quanto às complexidades e ignora-se as demais.

**d)** Número de elementos a ser processado.

**e)** Ordem das funções conforme respectivos crescimentos – CAS, Ω, Θ.

**f)** Chama-se de Cota Assintótica Superior (CAS) uma função que cresce mais rapidamente que outra a partir de determinado momento.

**8.**

O limite assintótico inferior representa o número mínimo de passos que o algoritmo irá executar para encontrar uma solução – o qual indica o melhor caso do algoritmo; já o limite assintótico superior mostra como se dá o crescimento de determinado algoritmo, representando o “número máximo” (conhecido no momento) de passos a serem executados no pior caso do algoritmo em questão.