

APS 03

Daniela Araldi

1.

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

b) $(2^0)+(2^1)+(2^2)+(2^3)+(2^4)+(2^5) = 63$

c) $(0 + 2^0) + (1 + 2^1) + (2 + 2^2) + (3 + 2^3) + (4 + 2^4) + (5 + 2^5) = 1 + 3 + 6 + 11 + 20 + 37 = 78$

d) $n + n + \dots + n = n * n = n^2$

2.

n	5n	4n ²	2 ⁿ⁻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 $4n^2$ é CAS para $5n$;

*Momento em que 2^{n-1} é CAS para $4n^2$.

3.

a) $2n^3 + n(n^2 - n - 1)$

$2n^3 + n^3 - n^2 - n$

Complexidade $O(n^3)$

b) $(n^2 + 2)/n$

$n^2/n + 2/n$

n

Complexidade $O(n)$

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

$(n^2 - n)/2$

Complexidade $O(n^2)$

d) $n + n^2 \log n$
Complexidade $O(n^2 \log n)$

4.

- a) L1. Θn
L2. $n \Theta(n) = O(n^2)$
L3. $n^2 n = O(n^3)$
L4. $O(1)$ // constante porque é uma atribuição normal

$$O(n) + O(n^2) + O(n^3) = \underline{O(n^3)}$$

- b) L1. $O(n)$
L2. $n O(n) = O(n^2)$
L3. $O(n^2)$
L4. $n^2 O(n) = O(n^3)$
L5. $O(n^3)$

$$O(n) + O(n^2) + O(n^2) + O(n^3) + O(n^3) = \underline{O(n^3)}$$

- 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. $n O(n) = O(n^2)$
L10. $n O(n^2) = O(n^3)$
L11. $O(1)$

$$(7 * O(1)) + (2 * O(n)) + O(n^2) + O(n^3) = \underline{O(n^3)}$$

5.

Para duas funções $g(n)$ e $f(n)$ temos que $f(n) = \Theta(g(n)) \leftrightarrow f(n) = \Omega(g(n))$ e $f(n) = O(g(n))$.

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

$f(n) = \Omega(g(n))$ 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 ($\Omega(n \log n)$).

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.