

Lista 3 de EDD1 - Questões Discursivas

1)

- a) $O(n + m)$
- b) $O(n^2)$
- c) $O(n \log_2 n)$

2)

Expressão	Termo Dominante	O()
$5 + 0.001n^5 + 0.025n$	$0.001n^5$	$O(n^5)$
$500n + 100n^{0.5} + 50n * \log(10, n)$	$100n^{0.5}$	$O(n^{0.5})$
$0.3n + 5n^{1.5} + 2.5n^{1.75}$	$2.5n^{1.75}$	$O(n^{1.75})$
$n^2 \log_2 n + n(\log_2 n)^2$	$n^2 \log_2 n$	$O(n^2 \log_2 n)$
$n \log_3 n + n \log_2 n$	$n \log_2 n$	$O(n \log_2 n)$
$3 \log_8 n + \log 2 \log 2 \log 2 n$	$3 \log_8 n$	$O(\log_8 n)$
$100n + 0.01n^2$	$0.01n^2$	$O(n^2)$
$0.01n + 100n^2$	$100n^2$	$O(n^2)$
$2n + n^{0.5} + 0.5n^{1.25}$	$0.5n^{1.25}$	$O(n^{1.25})$
$0.01n \log_2 n + n (\log_2 n)^2$	$n (\log_2 n)^2$	$O(n (\log_2 n)^2)$
$100n \log_3 n + n^3 + 100n$	n^3	$O(n^3)$
$0.003 \log_4 n + \log_2 \log_2 n$	$0.003 \log_4 n$	$O(\log_4 n)$

3)

- i) No código, através do loop iremos percorrer todo o vetor até que seja encontrado o número desejado ou até que cheguemos ao fim do vetor. Dessa forma, o pior caso desse código é quando o número desejado não faz parte do vetor, pois a quantidade de vezes que o loop irá rodar nessa situação será igual ao tamanho do vetor. Seguindo essa lógica, a complexidade do método é de $O(n)$ e n é o tamanho do vetor.
- ii) O código desse método possui a mesma complexidade do método i, só que dessa vez é utilizado o comando for ao invés do comando while. No caso de o número

desejado não estiver no vetor a complexidade também será $O(n)$ sendo n o tamanho do vetor, pois a quantidade de vezes que o for irá rodar será igual ao tamanho do vetor.

- iii) Nesse método iremos utilizar a busca binária para ver se o número faz parte de um vetor previamente ordenado. Na busca binária sempre dividimos o vetor na metade, se o desejado for menor que o número posicionado no meio olhamos para a primeira metade do vetor e continuamos dividindo na metade até que sobre um vetor com uma única posição. Nesse caso, o pior caso também será o caso onde o desejado não faz parte do vetor mas como nesse método estamos sempre dividindo o vetor no meio, a complexidade passa a ser $O(\log n)$.

4) Código

- 5) Nesse método temos 2 loops, o for externo irá apontar para uma posição do vetor a cada rodada enquanto o loop interno será utilizado para comparar a posição atual do vetor externo com o resto dos números presentes do vetor até chegar no final. Dessa forma, o crescimento assintótico do método é $O(n^2)$, resultado da multiplicação de n (quantidade de “rodadas” do loop externo) * n (quantidade de “rodadas” do loop interno).