

Aluno: Igor Bastos Albuquerque

Questão 1 -

A) A complexidade temporal é dada pelo número de operações a serem executadas pelo algoritmo relacionado a sua entrada. E trazendo isso para o nosso problema, temos que o pior caso é quando precisamos percorrer todo o vetor, chamando a função para cada elemento do vetor, n vezes

$$T(n) = 1 + n = O(n)$$

B) É a mesma, uma vez que na versão iterativa, precisamos de um for que depende de $f(n)$ e que dentro dele não há nenhuma outra dependências, apenas temos de complexidade constante, logo quem dita a complexidade é o $f(n)$ com uma complexidade no pior caso equivalente ao tamanho do vetor $O(n)$

Questão 2 -

A) Aparentemente a complexidade para ambas as construções é a mesma, uma vez que o tamanho do vetor (N) ao longo da execução do programa vão diminuindo pela metade resultando em um crescimento logarítmico, portanto são iguais.

B) A complexidade espacial da busca binária iterativa é $O(1)$. Isso ocorre porque a busca binária iterativa não requer alocação de memória adicional além de algumas variáveis locais. Já a complexidade temporal, é $O(\log(n))$

Questão 3 -

A) Tanto no pior quanto no melhor caso, há necessidade de se percorrer todo o vetor, que possui o tamanho N , a fim de verificar se está ou não ordenado, dessa forma, fazendo com que a complexidade seja $O(n)$

Questão 4 -

A) A complexidade do algoritmo no pior caso é $O(n)$ em que n é a posição do termo da sequência que queremos seja calculado. A comparação entre as versões iterativas e recursivas, mostram que a recursiva tem uma complexidade exponencial $O(2^n)$ em virtude das chamadas recursivas que

B)