

Centro Universitário Senac
Bacharelado em Ciência da Computação
Análise e projeto de algoritmos - ADO 01

Professor: Leonardo Takuno
{leonardo.takuno@gmail.com}

18 de fevereiro de 2021

1. Analise a complexidade de algoritmos iterativos a seguir. Para cada item, expresse a complexidade em termos de notação O , justifique a sua resposta:

- (a) `soma = 0;`
 `for(i = 1; i <= n; i++)`
 `for(j = 1; j <= i; j++)`
 `soma++;`
- (b) `soma = 0;`
 `for(i = 1; i <= n; i++)`
 `for(j = i; j <= n; j++)`
 `for(k = 1; k <= 100; k++)`
 `soma++;`
- (c) `soma = 0;`
 `for(i = n/2; i <= n; i++)`
 `for(j = i; j <= n/2; j++)`
 `for(k = 1; k <= n; k *= 2)`
 `soma++;`
- (d) `soma = 0;`
 `for(i = n/2; i <= n; i++)`
 `for(j = i; j <= n; j *= 2)`
 `for(k = 1; k <= n; k *= 2)`
 `soma++;`
- (e) `soma = 0;`
 `for(i = n*n; i <= n*n*n; i++)`
 `soma++;`

2. Considere o **problema de busca**:

Entrada: Uma sequência de n números $\langle a_1, a_2, \dots, a_n \rangle$ e um valor v .

Saída: Um índice i tal que $v = A[i]$ e $1 \leq i \leq n$, ou um valor -1 se v não aparece em A .

Escreva o pseudocódigo para a busca linear que busca v dentro da sequência.

3. Voltando ao problema de busca, observe que se a sequência A está ordenada, nós podemos comparar o meio da sequência com v e eliminar metade da sequência da busca. O algoritmo de **busca binária** repete este procedimento, dividindo ao meio a porção restante da sequência a cada passo. Escreva um pseudocódigo, ou iterativo ou recursivo, para a busca binária. Argumente que o tempo de execução de pior caso da busca binária é $O(\lg n)$

4. Considere a ordenação de n números armazenados em um vetor A utilizando a seguinte técnica: encontrar o menor elemento de A e trocá-lo com o primeiro elemento de A ; então, encontrar o segundo menor elemento de A e trocá-lo com o segundo elemento de A ; continuar dessa maneira até que os n elementos de A estejam ordenados. Escreva um pseudocódigo para este algoritmo, que é conhecido como **ordenação por seleção**. Forneça o tempo de execução do melhor e do pior caso da ordenação por seleção na notação θ .
5. Considere o algoritmo **INSERTION-SORT**(A) a seguir:

INSERTION-SORT(A)

```
1  for  $j \leftarrow 2$  to  $\text{length}[A]$ 
2      do  $\text{chave} \leftarrow A[j]$ 
3           $\triangleright$  Inserir  $A[j]$  na sequência ordenada  $A[1..j-1]$ .
4           $i \leftarrow j - 1$ 
5          while  $i > 0$  and  $A[i] > \text{chave}$ 
6              do  $A[i+1] \leftarrow A[i]$ 
7                   $i \leftarrow i - 1$ 
8           $A[i+1] \leftarrow \text{chave}$ 
```

Observe que o laço **enquanto** das linhas 5–7 do procedimento **INSERTION-SORT** utiliza uma busca linear (da direita para esquerda) no vetor $A[1..j-1]$. Será que, ao invés da busca linear, poderíamos utilizar uma busca binária para melhorar o tempo de execução do pior caso da ordenação por inserção para $\theta(n \lg n)$?