

Exercício 1 CAL

Análise da Complexidade do InsertSort

Kalyl Heings¹

¹Departamento de Ciências da Computação - Universidade do Estado de Santa Catarina (UDESC)
R. Paulo Malschitzki, 200 - Zona Industrial Norte, Joinville - SC, 89219-710

{andre.campos0702,leonardo.sell,kalyl.henigs}@edu.udesc.br

1. Algoritmo

```
1 void ordenaInsercao(int *v, int n)
2 {
3     int i, j, x;
4     for (i = 0; i < n; i++)
5     {
6         x = v[i];
7         j = i - 1;
8         while (j >= 0 && v[j] > x)
9         {
10            v[j + 1] = v[j];
11            j--;
12        }
13        v[j + 1] = x;
14    }
15 }
```

1.1. Questão 1

Descreva o funcionamento do algoritmo. Apresente as alterações efetuadas no vetor [50,40,10,30] passo a passo.

1.2. Questão 2

Faça uma análise da complexidade desse algoritmo em detalhes.

2. Resolução

2.1. Questão 1

O algoritmo funciona iterando sobre um vetor v recebido a partir da posição inicial $i = 0$, estabelecendo um pivô $x = v[i]$ nesta posição e uma variável auxiliar $j = i - 1$. Então, enquanto j for maior ou igual a 0 e o valor de v na posição j for maior que o valor de v na posição i , o algoritmo irá fazer $v[j + 1] = v[j]$ e então decrementará j em 1, isto é, irá trocar as posições dos valores no vetor e irá passar os valores menores para o início do vetor, para todos os valores do vetor enquanto for válido a condição descrita acima. Por fim, o pivô x é atribuído a nova posição $v[j + 1]$ e a iteração continuará até $i = n$, onde n é o tamanho do vetor.

Portanto, durante as iterações, o vetor sera:

- Iteração 1: $v = [50, 40, 10, 30]$
- Iteração 3: $v = [40, 50, 10, 30]$
- Iteração 1: $v = [10, 40, 50, 30]$
- Iteração 1: $v = [10, 30, 40, 50]$

2.2. Questão 2

Analisando o algoritmo linha a linha, para o pior caso:

- linha 3: $\mathcal{O}(1)$
- linha 4: $\mathcal{O}(n)$
- linha 6: $\mathcal{O}(1)$
- linha 7: $\mathcal{O}(1)$
- linha 8: $\mathcal{O}(n)$
- linha 10: $\mathcal{O}(1)$
- linha 11: $\mathcal{O}(1)$
- linha 13: $\mathcal{O}(1)$

Seguindo a estrutura do algoritmo, tem-se:

$$\mathcal{O}(1) + (\mathcal{O}(n) * \mathcal{O}(1) * \mathcal{O}(1) * (\mathcal{O}(n) * \mathcal{O}(1) * \mathcal{O}(1)))$$

Simplificando a expressão, tem-se que a complexidade final é $\mathcal{O}(n^2)$.