Algoritmos e estruturas de dados 1

Luiz Fernando de Freitas Oliveira 27 de março de 2021

Resolução da lista 1 de ordenação interna

1. Pois o algoritmo irá percorrer o vetor inteiro independente da ordem dos dados no vetor.

2.

3. INÍCIO

```
VARIÁVEIS:
i, j, temporária: INTEIRO;
lista[5] = 4,3,5,2,1: INTEIRO;
ESCREVA("Antes do sort:");
PARA i de 1 até 5 FAÇA:
     ESCREVA("Casa i da lista: lista[i]");
PARA i de 1 até 4 FAÇA:
     PARA j de i+1 até 5 FAÇA:
     SE(lista[j] ; lista[i]) FAÇA
          temporária ;- lista[j];
          lista[j] ;- lista[i];
          lista[i] ;- temporária;
ESCREVA("Depois do sort:");
PARA i de 1 até 5 FAÇA:
     ESCREVA("Casa i da lista: lista[i]");
FIM
```

```
A = \langle 10, 20, 3, 15, 5, 1, 4,
A = < 10, *3, 20, 15, 5, 1, 4,

A = < *3, 10, 20, 15, 5, 1, 4,

A = < 3, 10, *15, 20, 5, 1, 4,

A = < 3, 10, 15, *5, 20, 1, 4,
  = < 3, 10, *5, 15, 20, 1, 4,
  = < 3, *5, 10, 15, 20, 1, 4,
  = < 3, 5, 10, 15, *1, 20, 4,
  = < 3, 5, 10, *1, 15, 20, 4,
  = < 3, 5, *1, 10, 15, 20, 4,
  = < 3, *1, 5, 10, 15, 20, 4,
  = < *1, 3, 5, 10, 15, 20, 4,
    < 1, 3, 5, 10, 15, *4, 20,
    < 1, 3, 5, 10, *4, 15, 20,
     < 1, 3, 5, *4, 10, 15, 20, 2
< 1, 3, *4, 5, 10, 15, 20, 2
    < 1, 3, 4, 5, 10, 15, *2, 20
     < 1, 3, 4, 5, 10, *2, 15, 20
           3, 4, 5, *2, 10, 15, 20
          3, 4, *2, 5, 10, 15, 20
     < 1, 3, *2, 4, 5, 10, 15, 20
     < 1, *2, 3, 4, 5, 10, 15, 20
```

5.

- 6. Matematicamente, o algoritmo Insertion Sort se dá por: $\sum_{i=1}^{n-1} i = \frac{(n-1)\cdot n}{2} \cdot n.$ Logo, para 500 elementos teremos: $\frac{(500-1)\cdot 500}{2} = 124750 \text{ comparações.}$
- 7. O Insertion Sort é melhor que o Selection Sort em casos onde o vetor já está parcialmente ordenado.