**Ordenação por seleção**

Este método consiste em **selecionar** o menor valor do vetor e movê-lo para o primeiro índice, a seguir, seleciona-se o próximo menor valor, desconsiderando o já colocado na primeira posição.

A dinâmica do método de ordenação por seleção é sempre achar o menor valor do vetor e inseri-lo no começo, até completar todas as possibilidades de troca e ter o vetor ordenado.

Exemplo: 9 – 15 – 7 – 2 - 11 – 16

Simular em programa C e verificar o passo a passo executado.

Quantidade de passos: N-1, onde N é número de elementos.

**Ordenação por inserção**

Este método consiste em **inserir** cada um dos elementos em sua posição correta relativa à sequencia ordenada.

O vetor, neste método, é dividido em dois onde o primeiro contém os valores já ordenados e o segundo os valores a serem inseridos no primeiro. No início da ordenação considera-se que apenas o primeiro elemento do vetor pertence ao vetor já ordenado.

Exemplo: 9 – 15 – 7 – 2 - 11 – 16

Simular em programa C e verificar o passo a passo executado.

Quantidade de passos: N-1, onde N é número de elementos.

**Ordenação: método da bolha – Bubble Sort**

Este método consiste em comparar dois elementos consecutivos de um vetor e se o da esquerda é maior do que o da direita trocam de posição. Quando existem trocas, os elementos maiores tendem a deslocar-se para a direita e os menores para a esquerda.

Exemplo: 9 – 15 – 7 – 2 - 11 – 16

Simular em programa C e verificar o passo a passo executado.

Quantidade de passos: N-1, onde N é número de elementos.

Este método permite uma pequena melhoria, note que se em determinado passo não houver nenhuma troca é porque o vetor já está ordenado e neste caso o algoritmo pode parar sua execução.

**Comparações entre os algoritmos de ordenação**

Um dos conceitos que deve ser avaliado quanto aos algoritmos de ordenação é relacionado a sua estabilidade ou não. Um algoritmo de ordenação é dito estável quando elementos com mesmo valores não trocam de posição em relação à configuração original. Considere o algoritmo de seleção (que não é estável) e o seguinte vetor para exemplificação:

9(\*) – 9 (\*\*) – 4 – 3

Passo 1: 3 – 9 (\*\*) – 4 – 9(\*)

Passo 2: 3 – 4 – 9 (\*\*) – 9 (\*)

Passo 3: 3 – 4 – 9 (\*\*) – 9 (\*)

Nota: observe que o 9 (\*) estava a frente do 9 (\*\*) na configuração inicial. Porém ao final da execução do algoritmo de seleção note que o 9 (\*\*) ficou a frente do 9 (\*), caracterizando assim um algoritmo instável.

Outro ponto relevante é a análise do número de comparações efetuadas pelos algoritmos, sendo que devem ser avaliados o melhor caso, caso médio e o pior caso, sempre considerando a O (ordem). O objetivo principal aqui não é determinar o número de comparações exatos mas sim qual será o impacto no algoritmo a medida em que os números de elementos do vetor cresce.

Quadro Resumo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SELEÇÃO | INSERÇÃO | BOLHA (Bubble) – sem melhoria | BOLHA (Bubble) – com melhoria (troca) |
| Estavel x Instavel | Instável | Estável | Estável | Estável |
| Melhor caso | O(n2) | O(n) | O(n2) | O(n) - ordenado |
| Caso Médio | O(n2) | O(n2) | O(n2) | O(n2) |
| Pior Caso | O(n2) | O(n2) | O(n2) | O(n2) – inversamente ordenado |

**Exercícios**

1 – Efetuar o passo a passo segundo cada um dos algoritmos (seleção, inserção e bolha com melhoria) para ordenar os seguintes grupos de elementos. Informar a quantidade de passos necessários para ordenar cada um dos seguintes grupos.

1. 1 – 10 – 100
2. 9 – 5 – 3
3. 80 – 100 – 20 – 30 – 70 – 90 – 50 – 40 – 60 – 10