

Engenharia de Computação

Pesquisa e Classificação de Dados

Aula 04 – InsertionSort

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

Insertion Sort

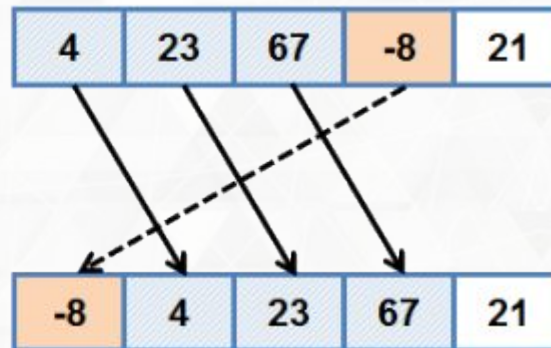
- Também conhecido como ordenação por inserção
 - Similar a ordenação de cartas de baralho com as mãos
 - Pegue uma carta de cada vez e a insira em seu devido lugar, sempre deixando as cartas da mão em ordem



Insertion Sort

- Funcionamento
 - O algoritmo percorre o array e para cada posição X verifica se o seu valor está na posição correta
 - Isso é feito andando para o começo do array a partir da posição X e movimentando uma posição para frente os valores que são maiores do que o valor da posição X
 - Desse modo, teremos uma posição livre para inserir o valor da posição X em seu devido lugar

Move as cartas maiores
para frente e insere na
posição vaga

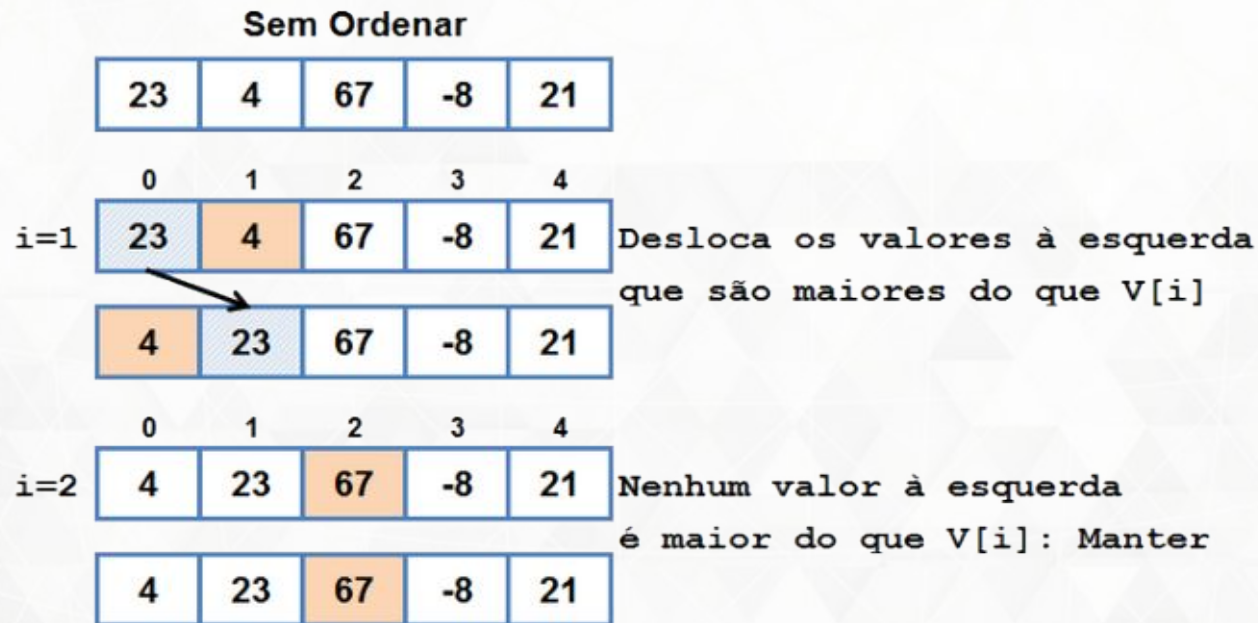


InsertionSort - Pseudocódigo

- InsertionSort (V, n)
 - i, j, eleito
 - PARA i <- 1 ATÉ (n-1) FAÇA
 - eleito <- A[i];
 - j ← i-1;
 - ENQUANTO ((j ≥ 0) E (eleito < A[j])) FAÇA
 - A[j+1] ← A[j];
 - j ← j-1;
 - A[j+1] ← eleito;

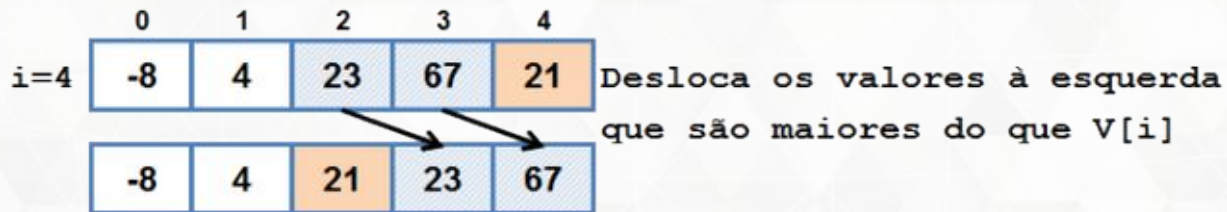
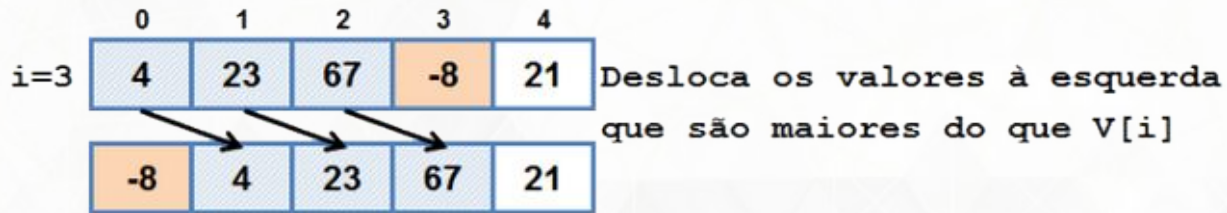
Insertion Sort - Passo a passo

- Para cada posição i , movimenta os valores maiores uma posição para frente no array



Insertion Sort – Passo a passo

- Para cada posição i , movimenta os valores maiores uma posição para frente no array



Ordenado

| | | | | |
|----|---|----|----|----|
| -8 | 4 | 21 | 23 | 67 |
|----|---|----|----|----|

Insertion Sort - Vantagens

- Fácil implementação
- Na prática, é mais eficiente que a maioria dos algoritmos de ordem quadrática. Como o SelectionSort e o BubbleSort.
- Um dos mais rápidos algoritmos de ordenação para conjuntos pequenos de dados. Superando inclusive o QuickSort
- **Estável:** não altera a ordem dos dados iguais
- **Online:** Pode ordenar elementos a medida que os recebe
 - Não precisa ter todo o conjunto de dados para colocá-los em ordem

Insertion Sort - Complexidade

- Considerando um array com n elementos, o tempo de execução é:
 - $O(n)$, melhor caso: os elementos já estão ordenados.
 - $O(n^2)$, pior caso: os elementos estão ordenados na ordem inversa.
 - $O(n^2)$, caso médio.

Exercício

- **1)** Simule o algoritmo utilizando cartas de um baralho ou uno.
- **2)** Implemente o **InsertionSort** em C considerando a seguinte assinatura de função

```
/**  
 * \brief Ordena o vetor usando InsertionSort  
 *  
 * \param v vetor a ser ordenado  
 * \param n tamanho do vetor  
 *  
 * Ordena o vetor usando o método InsertionSort  
 * Esse algoritmo tem um comportamento assintótico  $O(n^2)$   
 */  
void insertionSort(int *v, int n);
```