INSTITUTO FEDERAL
RIO GRANDE DO SUL

Estrutura de Dados II

Algoritmos de Ordenação:

- Ordenação por Inserção (Insert Sort)
- Ordenação por Seleção (Selection Sort)

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

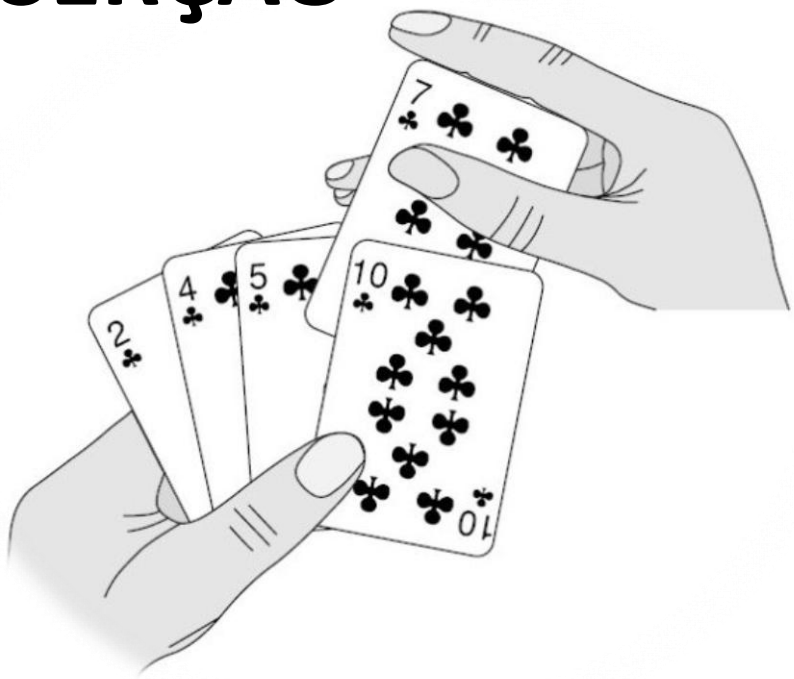
Algoritmos de Ordenação

Alguns Algoritmos de Ordenação:

- Bubblesort
- Ordenação por Contagem

- **Ordenação por Inserção**
- **Ordenação por Seleção**
  - Mergesort
  - Quicksort

ORDENAÇÃO POR INSERÇÃO



Ordenação por Inserção

É um método que percorre um vetor de elementos da esquerda para a direita e à medida que avança vai ordenando os elementos à esquerda.

Possui complexidade O(n) no melhor caso e $O(n^2)$ no caso médio e pior caso.

O funcionamento do algoritmo consiste em cada passo a partir do segundo elemento selecionar o próximo item da sequência e colocá-lo no local exato de acordo com o critério de ordenação.

Algoritmo de Ordenação por Inserção

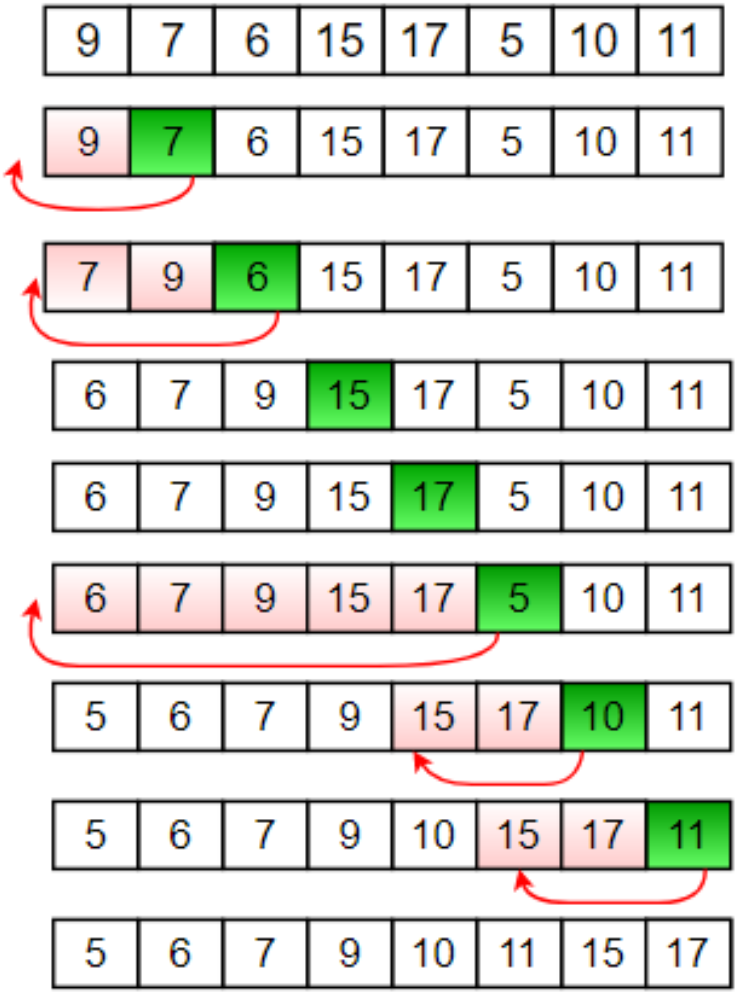
É um algoritmo simples, eficiente quando aplicado a um pequeno número de elementos.

Seu funcionamento é de percorrer um vetor de elementos da esquerda para a direita e à medida que avança vai deixando os elementos mais à esquerda ordenados.

Algoritmo de Ordenação por Inserção

- É possível comparar o ***Insertion Sort*** com o modo como de organizar cartas de um baralho.
- Imagine as cartas na mão e elas estão ordenadas. Você recebe uma nova carta, deve inserir na posição correta das cartas, de forma a que as cartas obedeçam à ordenação.
- A cada nova carta adicionada, a nova carta pode ser menor que algumas das cartas que você já tem na mão ou maior, e assim, você começa a comparar a nova carta com todas as cartas na sua mão até encontrar sua posição correta.
- Você insere a nova carta na posição correta. Então, você recebe outra carta e repete o mesmo procedimento, sucessivamente até a última.

Ordenação por Inserção



Algoritmo de Ordenação por Inserção

```
#include <stdio.h>
#include <stdlib.h>

main ()  {
    //Inicializaçã do Vetor
    int v[5]= {1,7,4,3,5} n=5;
    int i, j, chave;

    //Algoritmo de Ordenação
        for(j=1; j<n; j++)
        {
                chave = v[j];
                i = j-1;
                while(i >= 0 && v[i] > chave) {
                        v[i+1] = v[i];
                        i--;}
                        v[i+1] = chave;}

    //Laço de impressão do Vetor
        for (int q=0; q<5; q++)   {
                printf("%d \n",v[q]);
        }
}
```

Algoritmo de Ordenação por Inserção

- Quantas **iterações** são necessárias para ordenar um conjunto de 5 elementos: {1,3,5,9,10}?

- Quantas **iterações** são necessárias para ordenar um conjunto de 5 elementos: {10,9,5,3,1}?

ORDENAÇÃO POR SELEÇÃO

Algoritmo de Ordenação por Seleção

O ***selection sort*** compara a cada interação um elemento com os outros, visando encontrar o menor.

Dessa forma, podemos entender que não existe um melhor caso mesmo que o vetor esteja ordenado ou em ordem inversa serão executados.

A complexidade deste algoritmo será sempre $O(n^2)$.

Algoritmo de Ordenação por Seleção

Seu funcionamento consiste em se passar sempre o menor valor do conjunto para a primeira posição (ou o maior dependendo da ordem requerida), depois o de segundo menor valor para a segunda posição, e assim é feito sucessivamente com os (n-1) elementos restantes, até os últimos dois elementos.

Ele é um algoritmo simples de ser implementado em comparação aos demais **(sem recursividade).**

Não necessita de um vetor auxiliar, ocupa menos memória.

Ele é uns dos mais velozes na ordenação para um conjunto de dados pequenos.

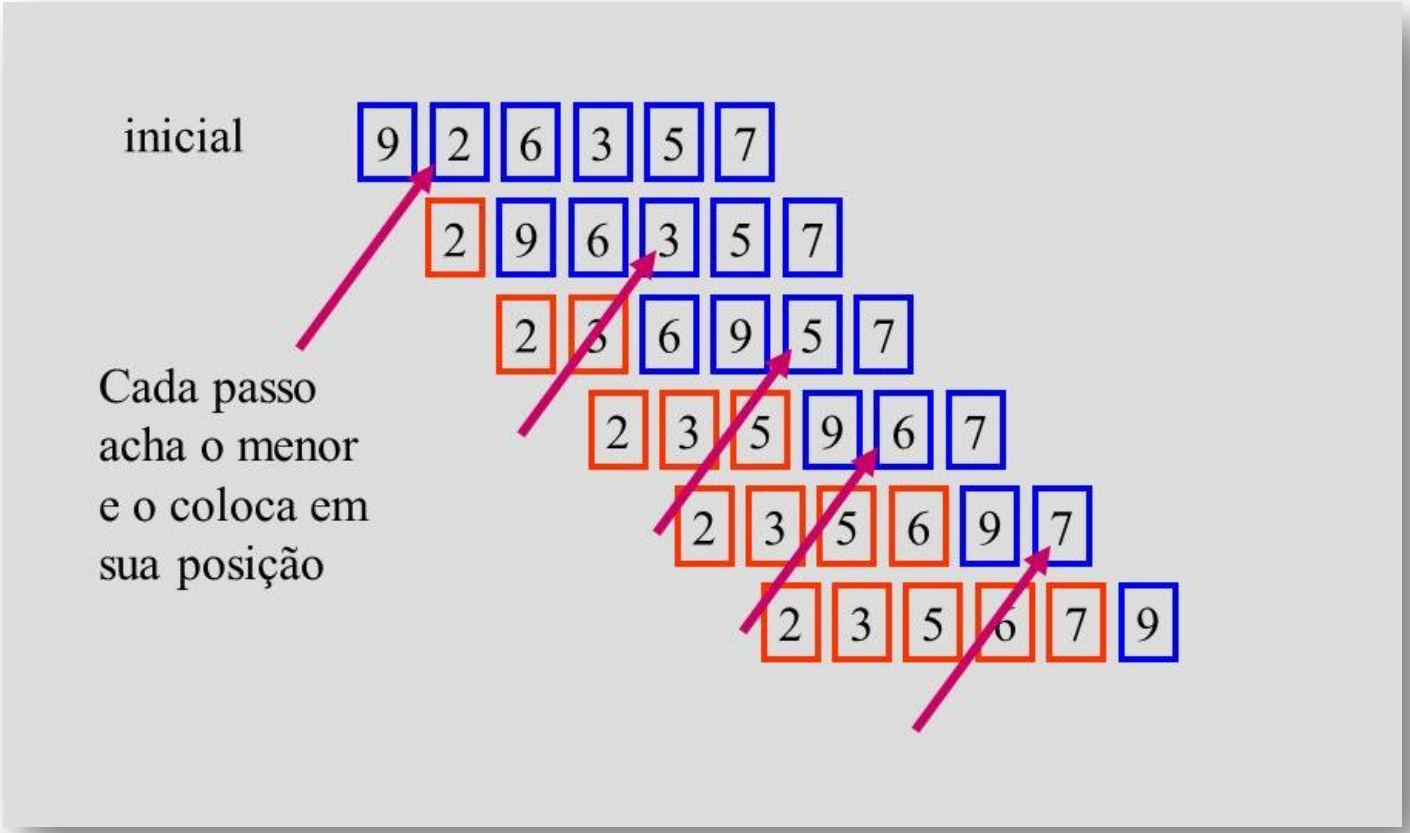
Algoritmo de Ordenação por Seleção

O Funcionamento do algoritmo de O**rdenação por Seleção** é o executado para um Conjunto de N números, os seguintes passos:

a) A cada iteração do algoritmo, selecione o menor elemento do conjunto deixe na sua posição correta final.

b) Resolva o problema para os elementos restantes.

Ordenação por Seleção



Algoritmo de Ordenação por Seleção

```
...

main () {
 //Inicialização do Vetor
    int num[5]= {1,7,4,3,5}, tam=5;

//Algoritmo de Ordenação
  int i, j, min;
  for (i = 0; i < (tam-1); i++) {
    min = i;
    for (j = (i+1); j < tam; j++) {
      if(num[j] < num[min]) {
        min = j;
      }}
    if (i != min) {
      int swap = num[i];
      num[i] = num[min];
      num[min] = swap;}}

//Laço de impressão do Vetor
for (int q=0; q<5; q++)
{
        printf("%d \n",num[q]);    }
}
```

Algoritmo de Ordenação Seleção

Qual é a diferença entre o Algoritmo de Seleção e o Algoritmo de Inserção?

```c
#include <time.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
main ()
{
    //Inicializaçã do Vetor
    FILE *txt;
    txt = fopen("Arquivo.txt", "r");
    float v[100000], n=100000;
    float j = 0,  aux;
    long int  i=0;

    printf("\nCarregando o Arquivo no Vetor");
    while (i< n)
    {
        fscanf(txt, "%f",&v[i]);
        i++;
    }
    fclose(txt);

    //Algoritmo de Ordenação
    //aqui

    printf("\nImprimindo Vetor Ordenado");
    txt = fopen("Arquivo_ordenado.txt", "w");

    //Laço de impressão do Vetor
    for (i=0; i<n; i++)
    {
        fprintf(txt, "%.0f\n",v[i]);
    }
    fclose(txt);
 }
```