EDB-I

Estrutura de Dados Básicas I

Aula 10
Algoritmos de Ordenação
Seleção e Inserção

(material baseado nas notas de aula do Prof. César Rennó Costa e Prof. Eiji Adachi)

UFRN UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

IMD INSTITUTO METRÓPOLE DIGITAL

Permutação

Uma permutação de um conjunto finito A é uma disposição dos elementos de A em uma sequência

Ex.: Dado o conjunto A ao lado, são permutações de A:

P1 = {1, 4, 9}

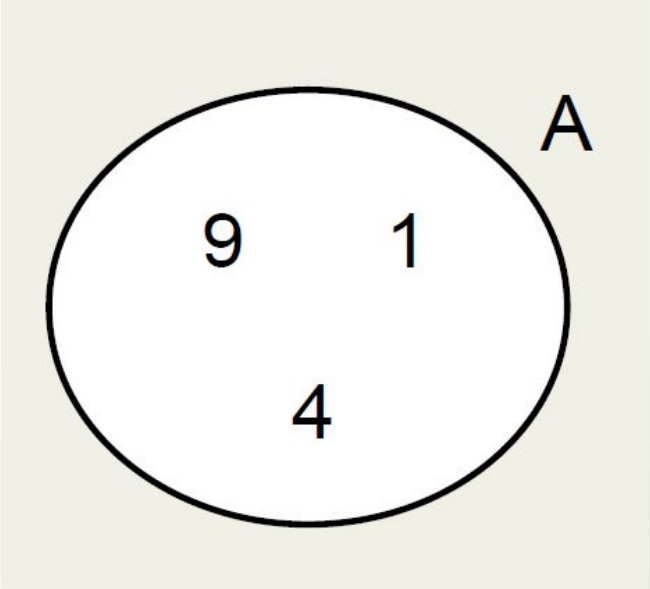
P2 = {1, 9, 4}

P3 = {4, 1, 9}

P4 = {4, 9, 1}

P5 = {9, 1, 4}

P6 = {9, 4, 1}



Ordenação

Para ordenar, é necessário estabelecer uma relação de ordem

Entrada

A: Coleção de elementos E_1, E_2, E_3, ... E_n

R: Relação de ordem sobre os elementos de A

Saída

Permutação A’ tal que elementos subsequentes de A’ obedeçam a relação de ordem R

Ordenação

Ex. Ordenação de inteiros

Entrada

A: Coleção de inteiros [2, 5, 1, 9, 10, 7]

R: Relação “menor que“ (<)

Saída

A’ = [1, 2, 5, 7, 9, 10]

Ordenação

Ex. Ordenação de inteiros

Entrada

A: Coleção de inteiros [2, 5, 1, 9, 10, 7]

R: Relação “maior que“ (>)

Saída

- A’ = [10, 9, 7, 5, 2, 1]

Ordenação

Ex. Ordenação de strings

Entrada

A: Coleção de strings[“beta”, “alfa”, “delta”, “charlie”]

R: Relação “ordem alfabética”

Saída

A’ = [“alfa”, “beta”, “charlie”, “delta”]

Ordenação

Qual seria uma estratégia de ordenação muito ruim?

Ordenação

```
Ordene( V[], N ) :
   Gere todas as possíveis permutações de V
   PARA CADA permutação P, FAÇA:
      Verifique se P está ordenada
      SE P está ordenada, ENTÃO:
         V = P
         RETORNE
      FIM_SE
   FIM_PARA
FIM
```

Complexidade N!

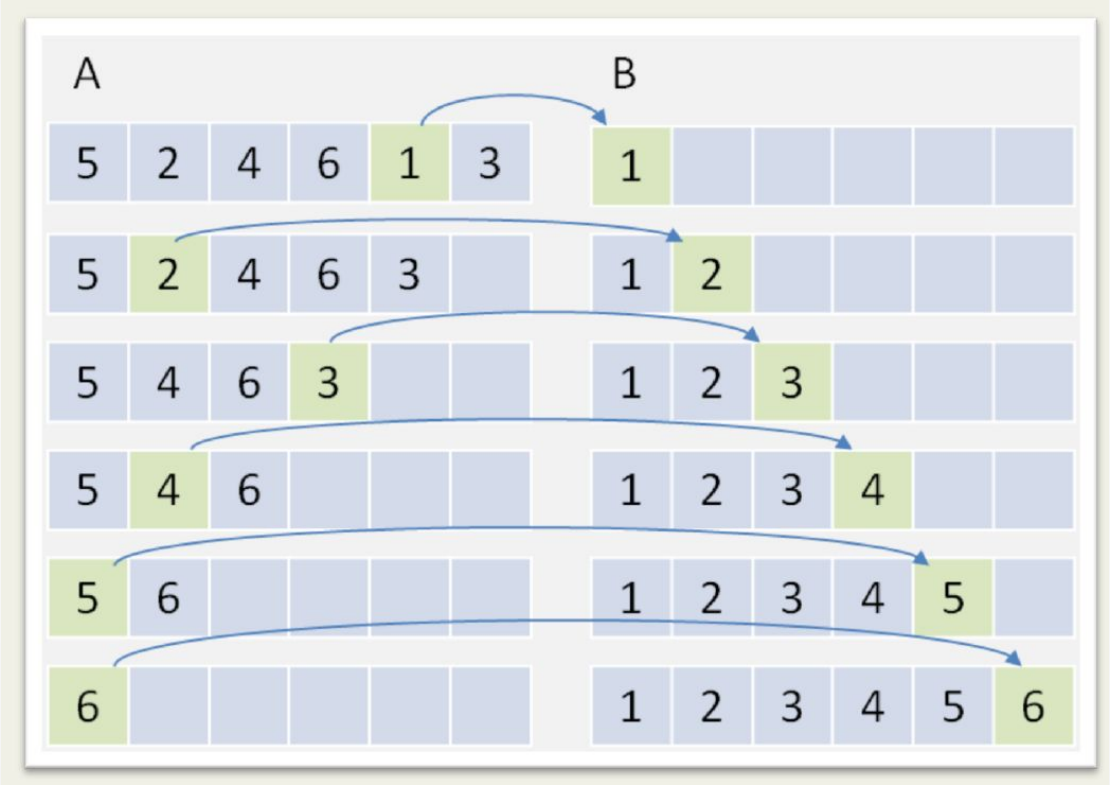
Ordenação por Seleção

Ideia geral

Dado um vetor de entrada v[N]

- Selecione o 1º menor elemento e coloque-o na 1ª posição
- Selecione o 2º menor elemento e coloque-o na 2ª posição
- ...
- Selecione o N-ésimo menor elemento e coloque-o na N-ésima posição

Ordenação por Seleção



Ordenação por Seleção

Exemplo de ordenação crescente

1. inicie com o primeiro elemento do vetor
2. encontre o *menor* elemento a partir do elemento atual
3. troque o elemento encontrado com o elemento atual
4. repita a partir do passo 2 para o próximo elemento

Exemplo
v=[6,3,5,1,2]

Iteração 0
v=[6,3,5,1,2]
-elemento atual: 6 (índice 0)
-menor elemento: 1 (índice 3)
troca: v=[6,3,5,1,2] -> v=[1,3,5,6,2]

Iteração 1
v=[1,3,5,6,2]
-elemento atual: 3 (índice 1)
-menor elemento: 2 (índice 4)
troca: v=[1,3,5,6,2] -> v=[1,2,5,6,3]

Iteração 2
v=[1,2,5,6,3]
-elemento atual: 5 (índice 2)
-menor elemento: 3 (índice 4)
troca: v=[1,2,5,6,3] -> v=[1,2,3,6,5]

Iteração 3
v=[1,2,3,6,5]
-elemento atual: 6 (índice 3)
-menor elemento: 5 (índice 4)
troca: v=[1,2,3,6,5] -> v=[1,2,3,5,6]

Ordenação por Seleção

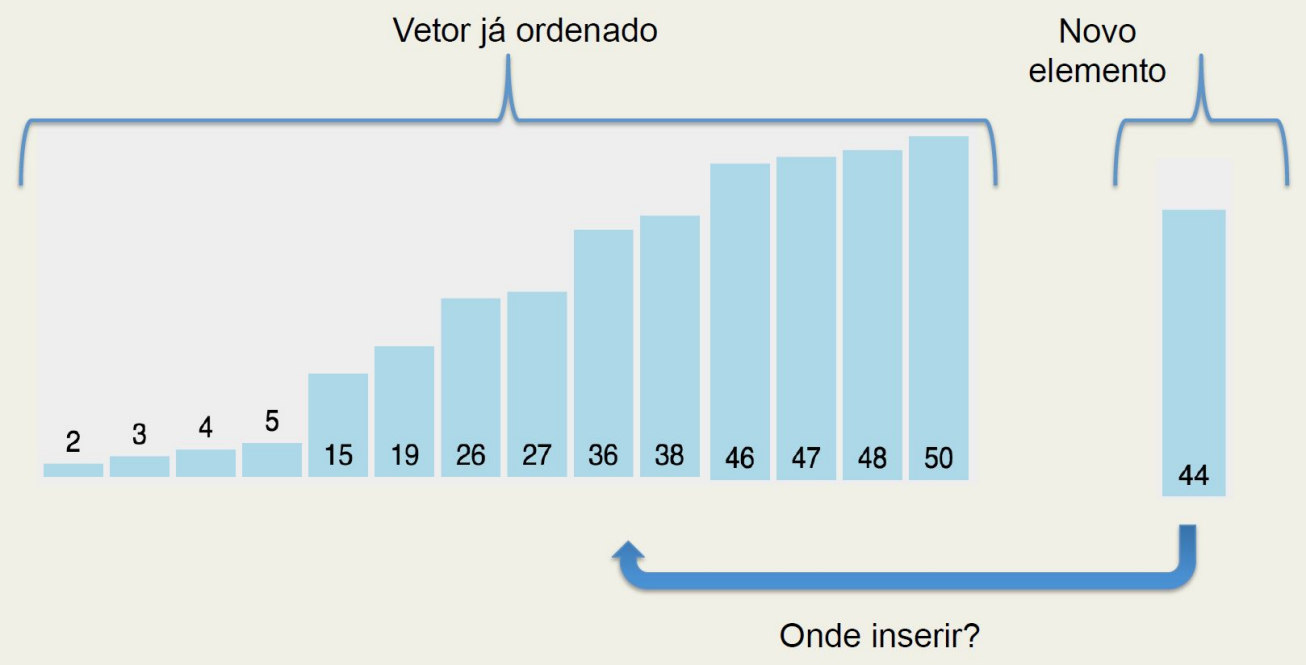
```
SELECTION-SORT(v[N])
1.    for j ← 0 to (N-2)
2.          indiceMenor ← j
3.          for i ← (j+1) to (N-1)
4.                if v[i] < v[indiceMenor]
5.                      indiceMenor ← i
6.           Troque v[ j ] ↔ v[indiceMenor]
```

Ordenação por Seleção

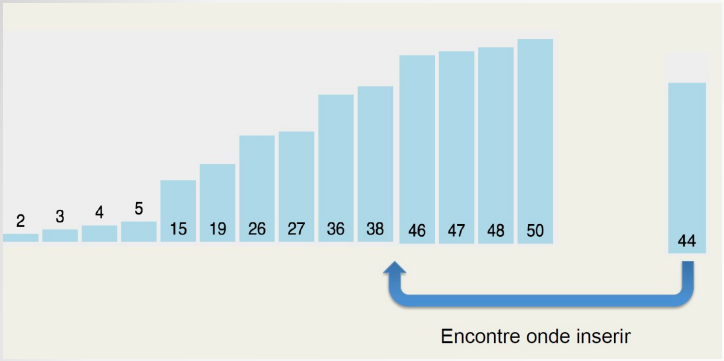
```
void SelectionSort(int vetor[], int tam) {
    for (int indice = 0; indice < tam; ++indice) {
        int indiceMenor = indice;
        for (int indiceSeguinte = indice+1; indiceSeguinte < tam; ++indiceSeguinte) {
            if (vetor[indiceSeguinte] < vetor[indiceMenor]) {
                indiceMenor = indiceSeguinte;
            }
        }
        int aux = vetor[indice];
        vetor[indice] = vetor[indiceMenor];
        vetor[indiceMenor] = aux;
    }
 }
```

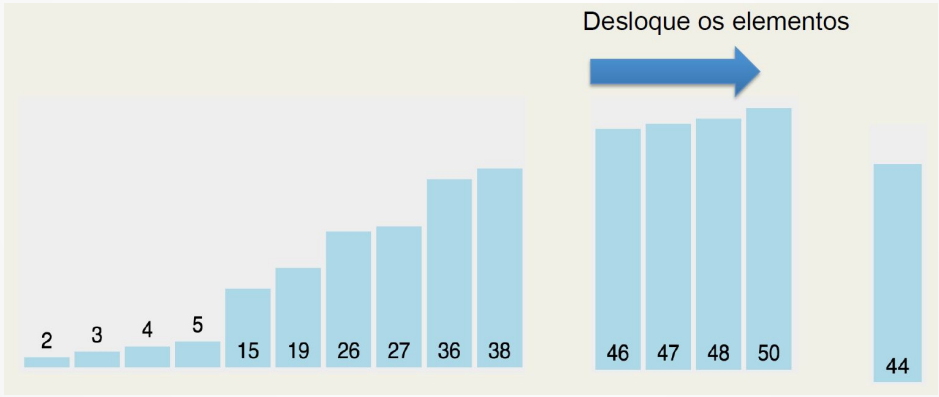
Exemplo
v=[6,3,5,1,2]

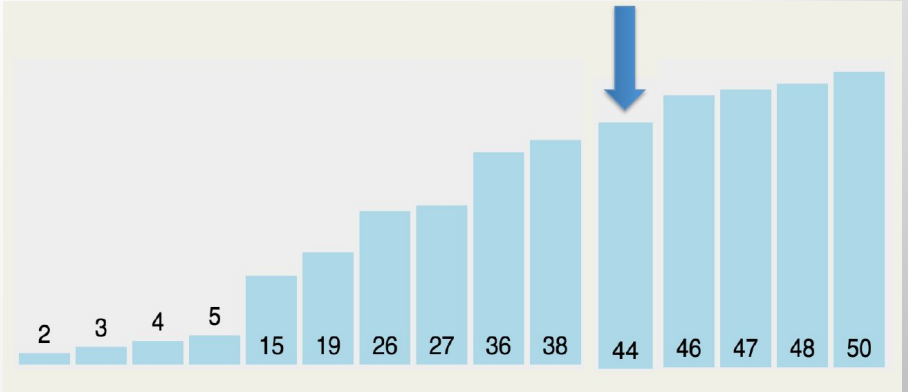
Ordenação por Inserção



Ordenação por Inserção







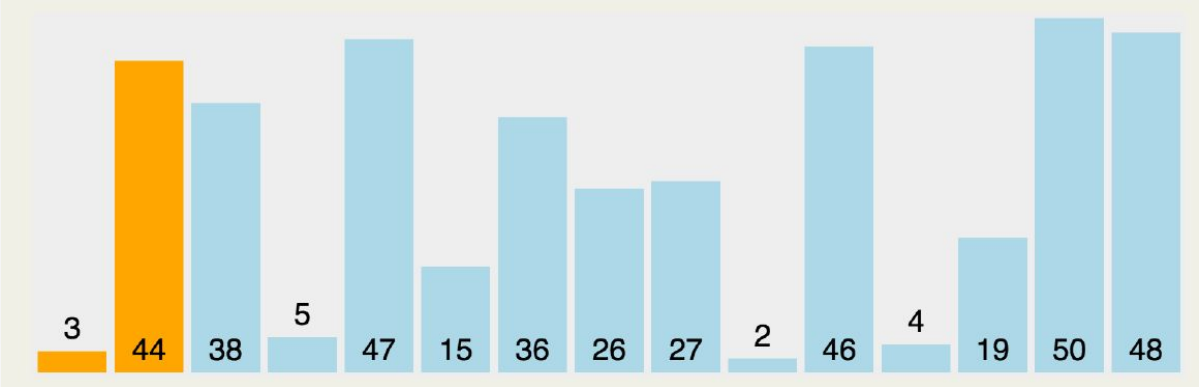
Ordenação por Inserção

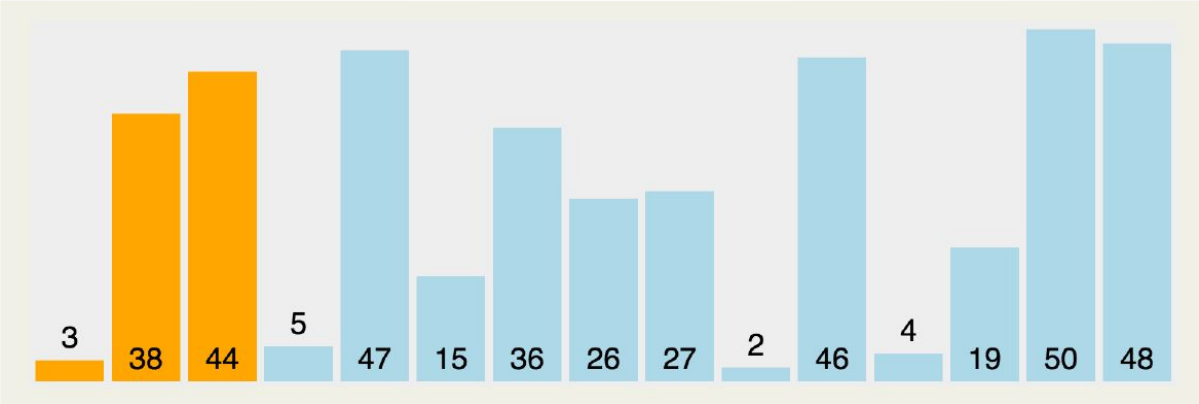
Idéia geral

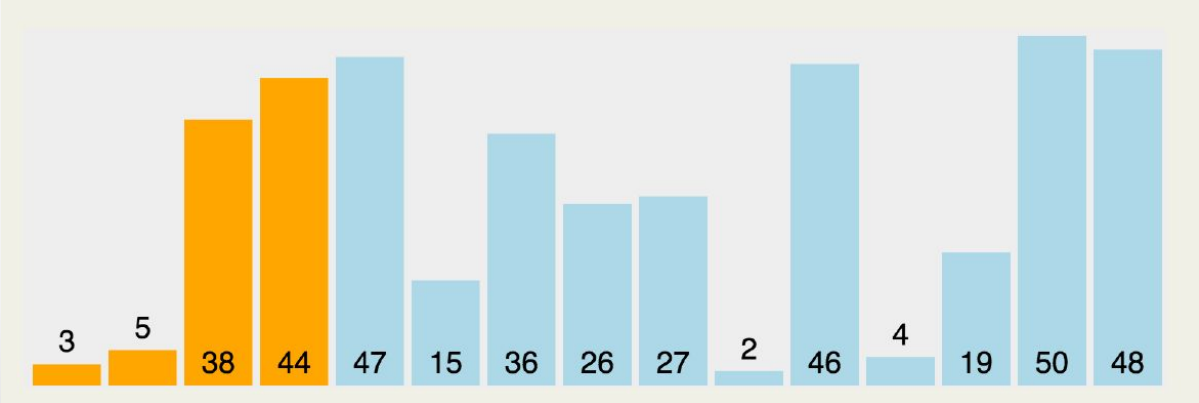
Dado um vetor v[N], ele terá uma parte ordenada e outra não ordenada

- Insira primeiro elemento da parte não ordenada na parte ordenada

Ordenação por Inserção







Ordenação por Inserção

```
INSERTION-SORT ( v[N] ):
 1.   for i ←1 to N
 2.         novo = v[i] // 'novo' é o primeiro não ordenado
 3.         j = i-1 // 'j' é o fim da parte ordenada
 4.         while j ≥ 0 && novo < v[j]
 5.               v[j+1] = v[j] // desloca elementos
 6.               j = j-1
 7.         end
 8.         v[j+1] = novo // insere novo elemento no local
 9.   end
```

Exemplo
v=[6,3,5,1,2]

Ordenação por Inserção

```
void insertion_sort(std::vector<int> &vetor) {

    for (int i = 1; i < vetor.size(); i++) {
       int escolhido = vetor[i];
       int j = i - 1;

       while ((j >= 0) && (vetor[j] > escolhido)) {
          vetor[j + 1] = vetor[j];
          j--;
       }

       vetor[j + 1] = escolhido;
    }
 }
```

Ordenação por Seleção
vs
Ordenação por Inserção

Ordenação por seleção

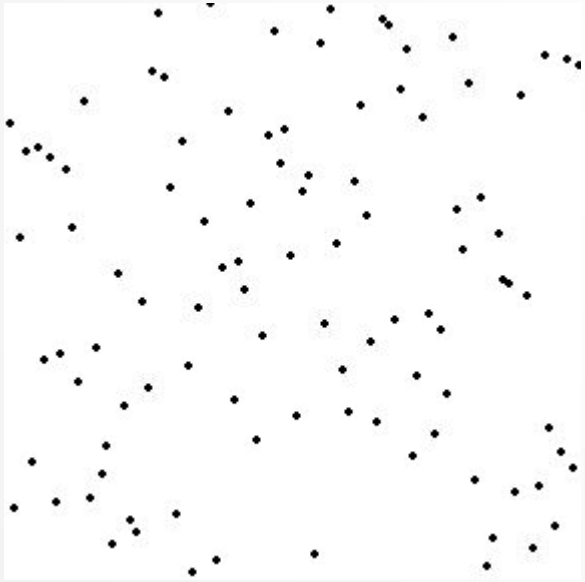
– Pior e melhor caso: $O(n^2)$
  - Qualquer vetor

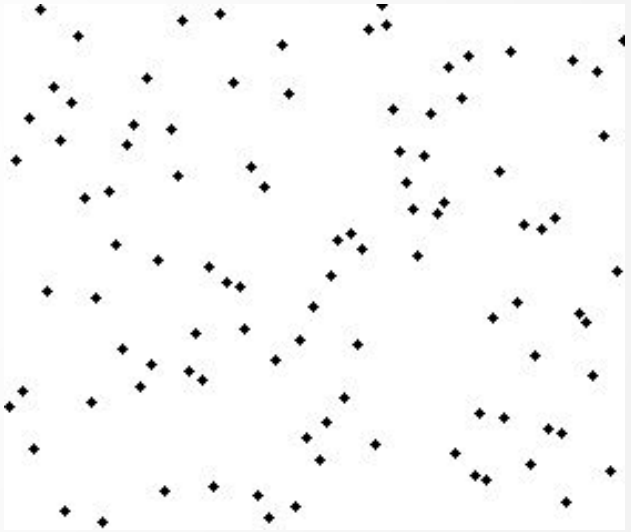
Ordenação por inserção

– Pior caso: $O(n^2)$
  - Vetor ordenado em ordem inversa

– Melhor caso: $O(n)$
  - Vetor já ordenado

Ordenação por Seleção
vs
Ordenação por Inserção





Ordenação por Seleção e Inserção

https://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html