

# Ordenação por seleção

## Programação de computadores II

Prof. Renan Augusto Starke

Instituto Federal de Santa Catarina – IFSC  
Campus Florianópolis  
`renan.starke@ifsc.edu.br`

24 de junho de 2016



**INSTITUTO FEDERAL**  
**SANTA CATARINA**

Ministério da Educação  
Secretaria de Educação Profissional e Tecnológica  
**INSTITUTO FEDERAL DE SANTA CATARINA**

# Tópicos da aula

- 1 Introdução
- 2 Ordenação por seleção
- 3 Exercícios

## 1 Introdução

## 2 Ordenação por seleção

## 3 Exercícios

- ▶ Entender alguns fundamentos matemáticos relacionados com algoritmos de ordenação
- ▶ Aprender as ordenações por seleção
- ▶ Conhecer o *Selection Sorting*
- ▶ Aplicar ordenação nas estruturas de dados conhecidas

1 Introdução

2 Ordenação por seleção

3 Exercícios

# Ordenação por seleção

## Ordenação por seleção

Compreende nos algoritmos onde a ordenação é realizada por *seleção* de elementos.

- ▶ Constrói-se a sequência ordenada, elemento por elemento, adicionando-os ordenadamente
- ▶ A cada passo, um elemento remanescente é **selecionado** integrar a nova sequência
- ▶ Elementos são sempre adicionados no final da sequência
- ▶ Esta característica que torna esta ordenação diferente da **Ordenação por inserção**

Algoritmos mais conhecidos:

- ▶ *Straight Selection Sorting*
- ▶ *Heap Sort*

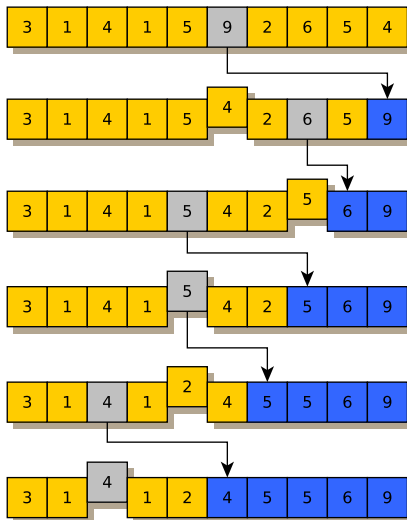
## Selection Sorting

O *Selection Sorting* é a variação mais simples dos algoritmos de ordenação por seleção. Pode-se ordenar selecionando o maior ou o menor elemento da sequência.

Para ordenar uma sequência  $S = \{s_1, s_2, \dots, s_n\}$ , com  $|S| > 1$ , o Selection Sorting realiza os seguintes passos:

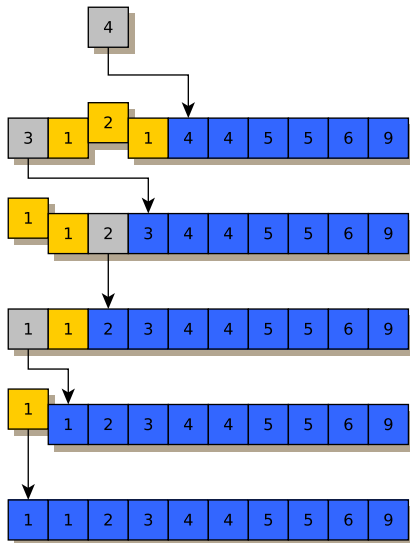
- 1 Busca-se o maior (ou menor) elemento dos remanescentes não ordenados.
- 2 Troca-se este elemento com o do final (ou inicial) da sequência.

# Selection Sorting

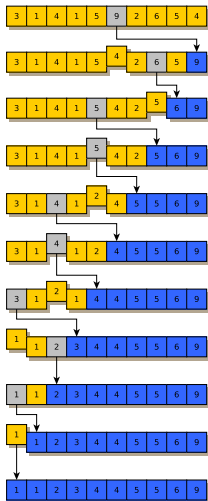




# Selection Sorting



# Selection Sorting



```
1 void select_sort(int *array, int n) {  
2  
3     for (i = n; i > 1; i--) {  
4  
5         int max = 0;  
6  
7         for (j = 1; j < i; j++)  
8             if (array[j] > array[max])  
9                 max = j;  
10  
11         swap (i - 1, max);  
12     }  
13 }
```

- Complexidade:  $O(n^2)$
- Esta ordenação é estável?

1 Introdução

2 Ordenação por seleção

3 Exercícios

- ▶ Implemente o *Selection sorting* para um vetor de inteiros.
- ▶ Estenda a implementação da lista duplamente encadeada com uma função de ordenação por *selection sorting*.