

# Insertionsort

Análise de Algoritmos – Ciência da Computação



Prof. Daniel Saad Nogueira  
Nunes

IFB – Instituto Federal de Brasília,  
Campus Taguatinga



# Sumário

---

## 1 Insertionsort



# Insertionsort

---

## Insertionsort

O projeto do algoritmo Insertionsort segue um argumento análogo à indução matemática.

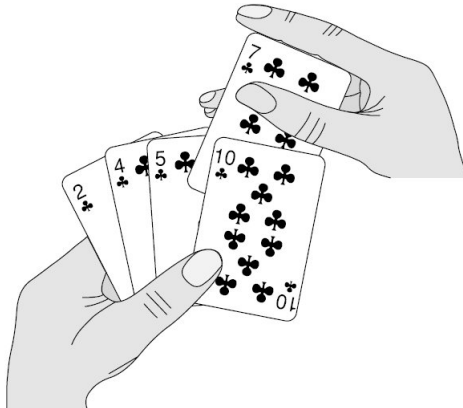
- Caso base: uma sequência com um elemento está ordenada.
- Passo de indução: a inserção de um elemento em uma sequência ordenada na posição correta também gera uma sequência ordenada.



# Insertionsort

---

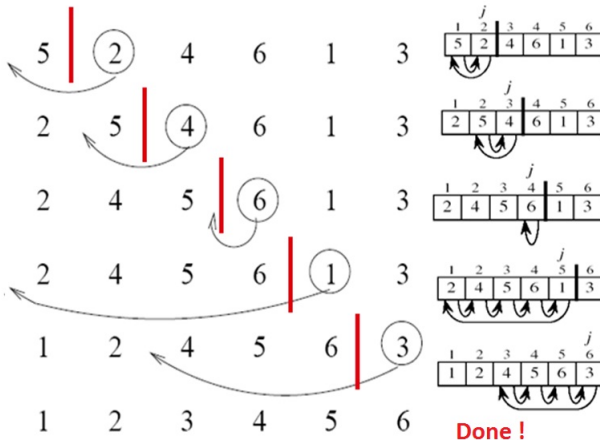
## Analogia com Baralho





# Insertionsort

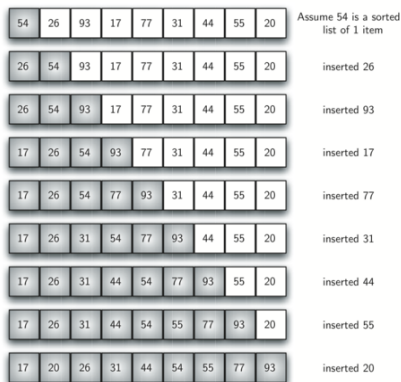
## Exemplo





# Insertionsort

## Exemplo





# Insertionsort

---

---

## Function Insertionsort

---

**Input:**  $V[0, n - 1]$

**Output:**  $V[0, n - 1]$ ,  $V[i] \leq V[i + 1], 0 \leq i < n - 1$

```
1 for(  $i \leftarrow 1; i < n; i++$  )
2    $chave \leftarrow V[i]$ 
3   for(  $j \leftarrow i - 1; j \geq 0 \wedge V[j] > chave; j--$  )
4      $V[j + 1] \leftarrow V[j]$ 
5    $V[j + 1] \leftarrow chave$ 
```

---



# Sumário

---

## 2 **Análise**





# Insertionsort

---

## Análise

- No pior caso, são necessários  $n - 1$  iterações sobre a sequência original. Na iteração  $i$  são realizadas no máximo,  $i$  comparações ao todo.
- A inserção do elemento na posição correta, também necessita de  $i$  operações de troca (em vetores). Portanto, o número de comparações do algoritmo (o mesmo número de trocas em vetores) é dado como:

$$\sum_{i=0}^{n-1} i = 1 + 2 + \dots + n - 1 \in \Theta(n^2)$$



# Insertionsort

---

## Observação

- Eficiente para entradas pequenas.
- Mais rápido na prática do que outros algoritmos quadráticos (como o Bubblesort).

In-place	Estável
✓	✓