



# Insertion Sort

Diogo Oliveira  
diogoespinharaoliveira@gmail.com



*A ideia é visualizar os elementos da lista como cartas de baralho.*

*O ponto chave é verificar se um elemento é menor do que seu antecessor.*

# Exemplo

<b>5</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>4</b>
----------	----------	----------	----------	----------

# Iteração 1

1

5	1	3	2	4
---	---	---	---	---

***$i - 1$***

***$i$***

***$j$***

***$1 < 5 ?$***

# Iteração 1

1

5	5	3	2	4
---	---	---	---	---

***$i - 1$***

***$i$***

***$j$***

***Joga 5 para a direita***

# Iteração 1

1

5	5	3	2	4
---	---	---	---	---

*i*

*j*

*Fim do laço i*

# Iteração 1

1

1	5	3	2	4
---	---	---	---	---

*i*

*j*

**Insira a chave (1) na posição i**

# Iteração 1

1

1	5	3	2	4
---	---	---	---	---

*j*

**Resultado da iteração 1**



## Iteração 2

3

1	5	3	2	4
---	---	---	---	---

***$i - 1$***

***$i$***

***$j$***

**$3 < 5 ?$**

## Iteração 2

3

1	5	5	2	4
---	---	---	---	---

*$i - 1$*

*$i$*

*$j$*

*Joga 5 para a direita*

## Iteração 2

3

1	5	5	2	4
---	---	---	---	---

***$i - 1$***

***$i$***

***$j$***

***$3 < 1 ?$***

## Iteração 2

3

1	5	5	2	4
---	---	---	---	---

*i*

*j*

*Fim do laço i*

## Iteração 2

3

1	3	5	2	4
---	---	---	---	---

***j***

***j***

***Insira a chave (3) na posição i***

## Iteração 2

3

1	3	5	2	4
---	---	---	---	---

*j*

***Resultado da iteração 2***

# Iteração 3

2

1	3	5	2	4
---	---	---	---	---

*$i - 1$*

*$i$*

*$j$*

*$2 < 5 ?$*

# Iteração 3

2

1	3	5	5	4
---	---	---	---	---

*$i - 1$*

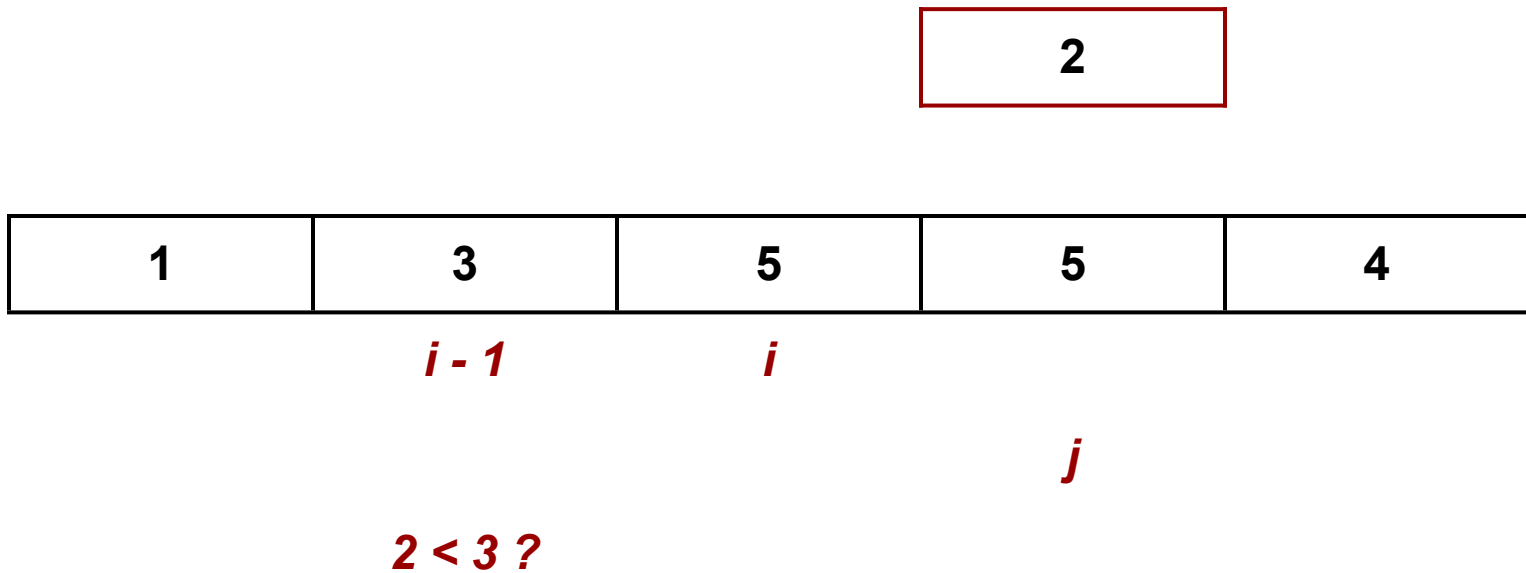
*$i$*

*$j$*

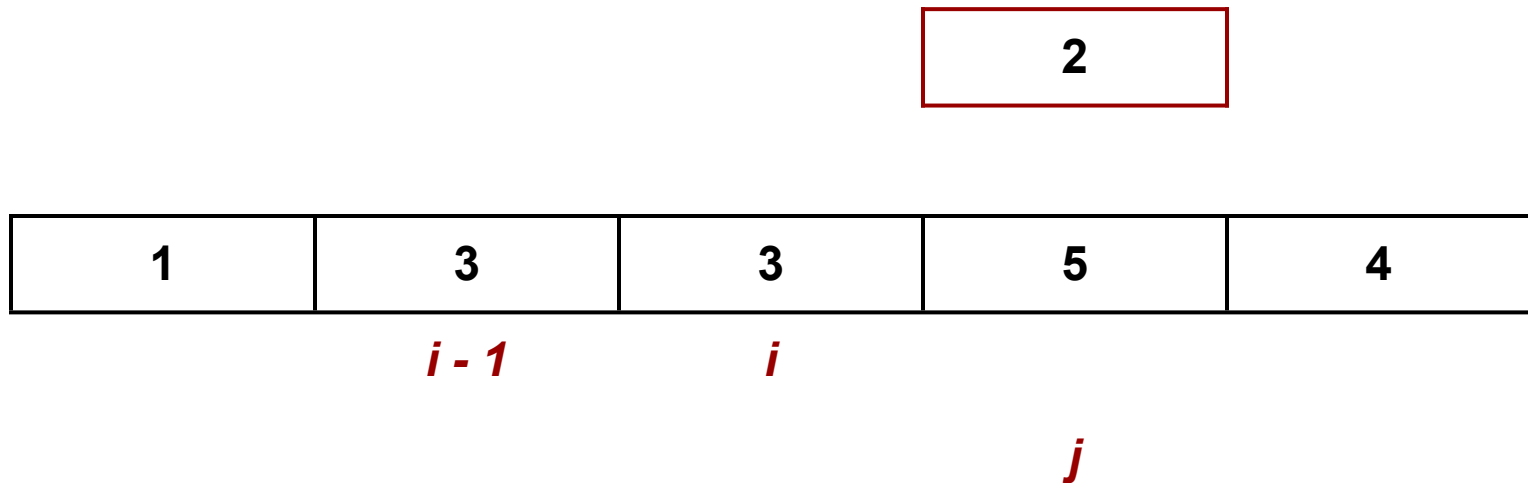
*Joga 5 para a direita*



# Iteração 3



# Iteração 3



*Joga 3 para a direita*

# Iteração 3

2

1	3	3	5	4
---	---	---	---	---

***$i - 1$***

***$i$***

***$j$***

***$2 < 1 ?$***

# Iteração 3

2

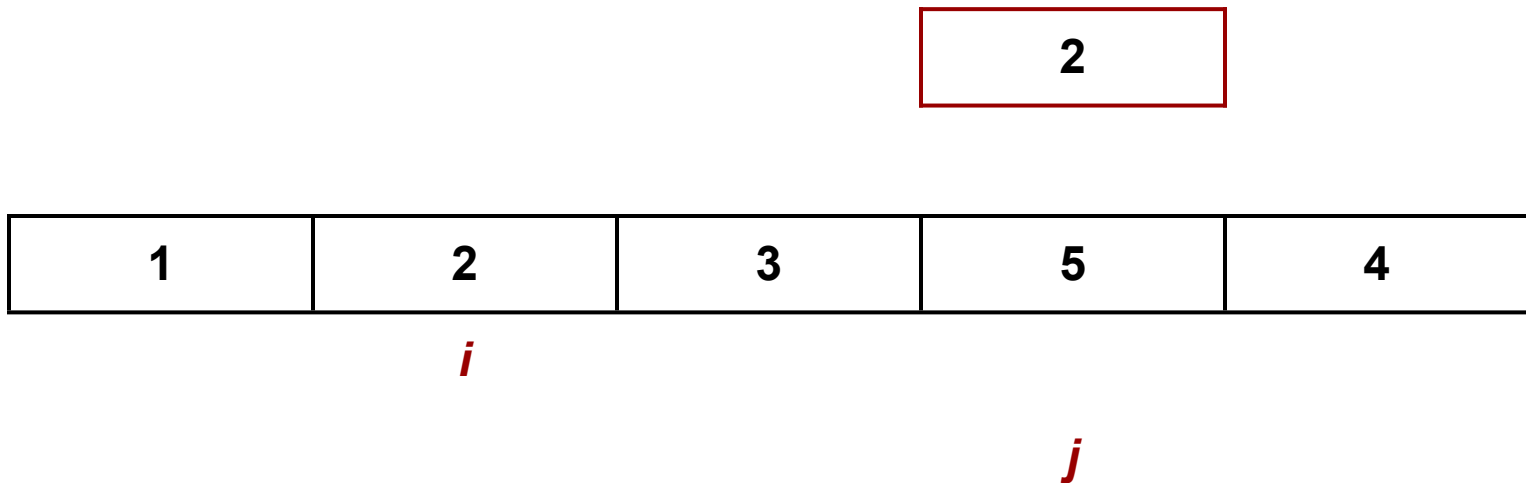
1	3	3	5	4
---	---	---	---	---

*i*

*j*

*Fim do laço i*

## Iteração 3



*Insira a chave (2) na posição  $i$*

# Iteração 3

2

1	2	3	5	4
---	---	---	---	---

*j*

***Resultado da iteração 3***

# Iteração 4

4

1	2	3	5	4
---	---	---	---	---

***$i - 1$***

***$i$***

***$j$***

***$4 < 5 ?$***

# Iteração 4

4

1	2	3	5	5
---	---	---	---	---

*$i - 1$*

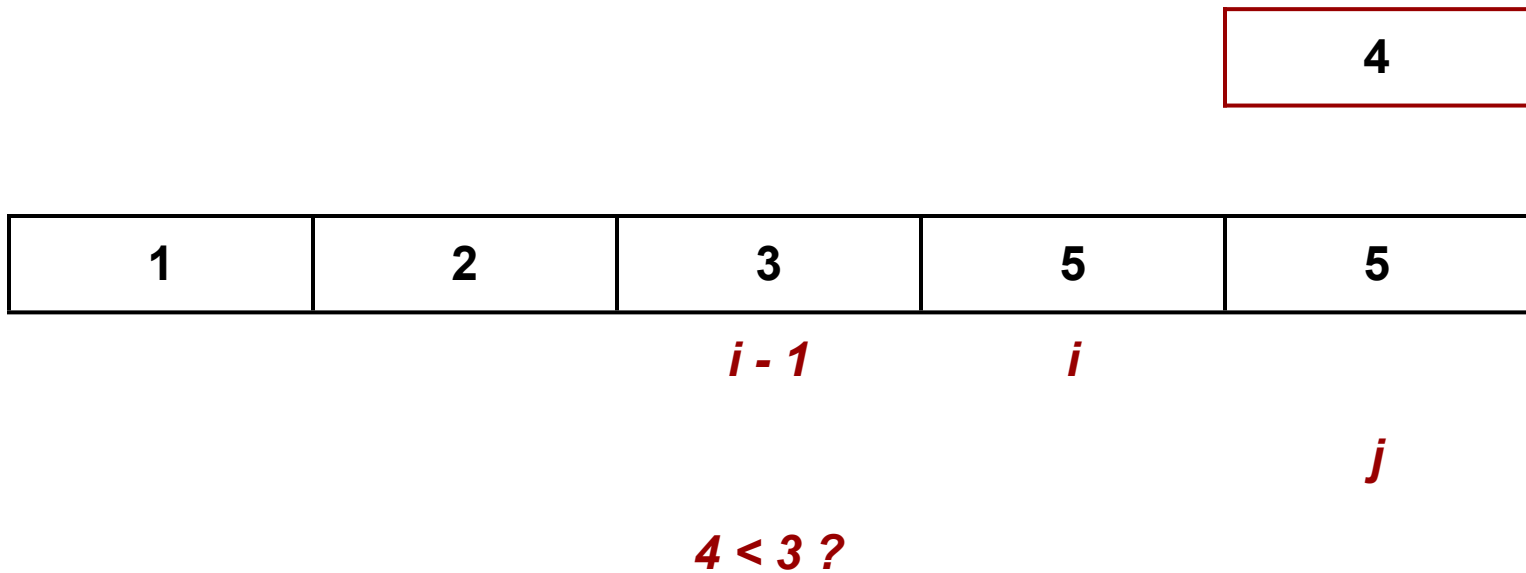
*$i$*

*$j$*

*Joga 5 para a direita*



# Iteração 4



# Iteração 4

4

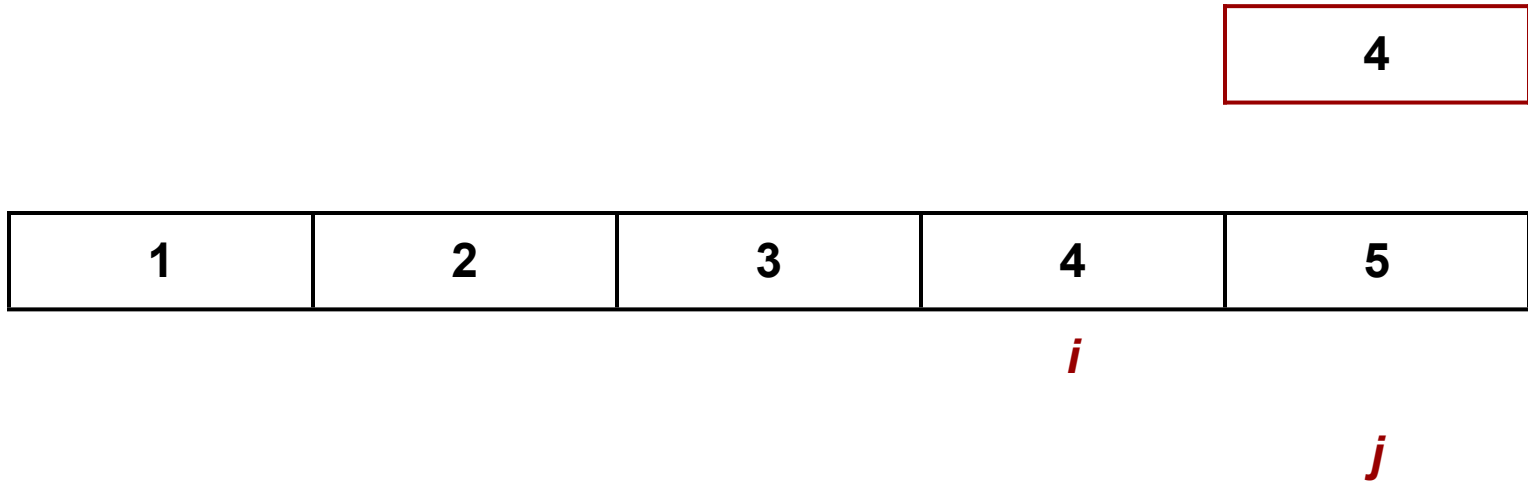
1	2	3	5	5
---	---	---	---	---

*i*

*j*

*Fim do laço i*

# Iteração 4



***Insira a chave (4) na posição *i****

# Iteração 4

4

1	2	3	4	5
---	---	---	---	---

*j*

***Resultado da iteração 4***

# Resultado final

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
----------	----------	----------	----------	----------



*Caso o valor da chave (carta) não seja menor do que seu antecessor ( $i - 1$ ), então o fluxo pula para a próxima iteração (carta).*

# Exercício

- ▶ Ordene as listas a seguir utilizando Insertion Sort;

<b>9</b>	<b>12</b>	<b>2</b>	<b>3</b>	<b>8</b>	<b>0</b>	<b>6</b>
----------	-----------	----------	----------	----------	----------	----------

<b>21</b>	<b>23</b>	<b>2</b>	<b>34</b>	<b>245</b>	<b>33</b>	<b>66</b>
-----------	-----------	----------	-----------	------------	-----------	-----------

# Implementação em sala





# Características

- ▷ Algoritmo simples;
  - Fácil de implementar e entender;
- ▷ In place e Stable
- ▷ Melhor desempenho que o Selection Sort;
- ▷ Online: não requer toda a entrada de início

# Laboratório22

- ▷ Implemente o algoritmo Insertion Sort para ordenar uma lista duplamente encadeada;