

Programação De Computadores

Professor : Yuri Frota

www.ic.uff.br/~yuri/prog.html

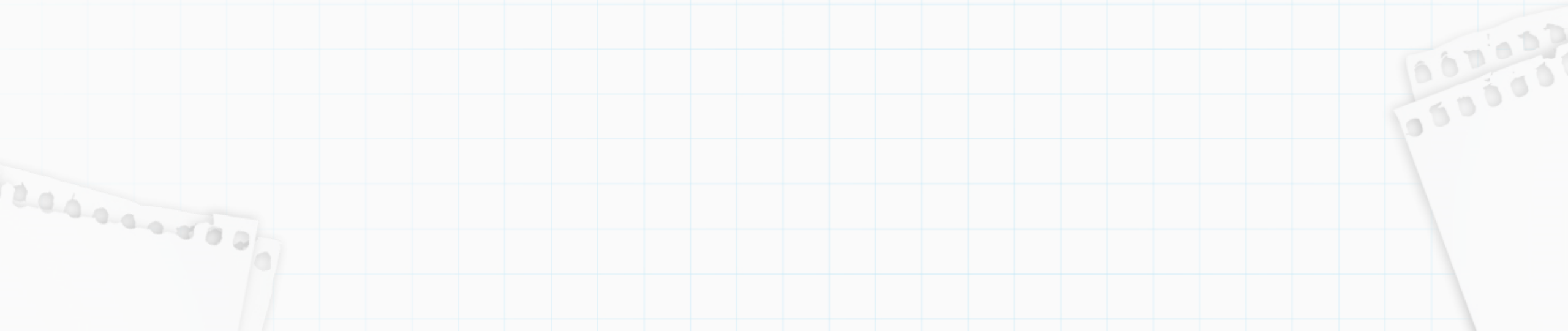
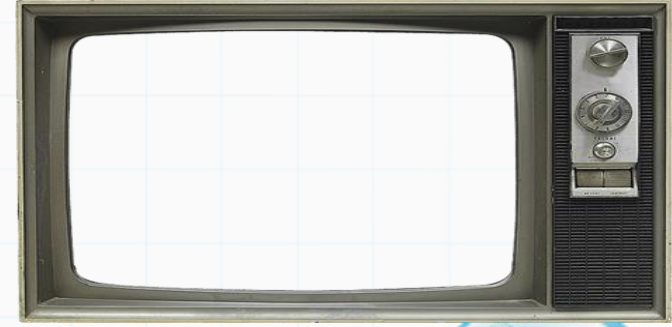
yuri@ic.uff.br



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

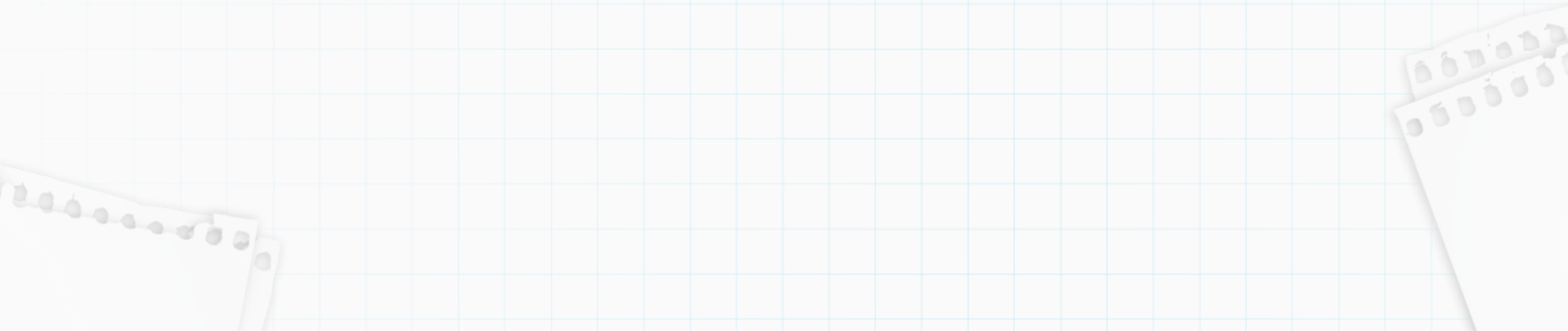
Vamos iterativamente olhar para as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

i

$i+1$



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente olhar para as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

i

$i+1$

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

i

$i+1$



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente olhar para as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

i

$i+1$

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

i

$i+1$

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

i

$i+1$



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

o vetor está ordenado ?
o vetor está mais ordenado ?



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

vamos repetir a varredura

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

i $i+1$

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

i $i+1$

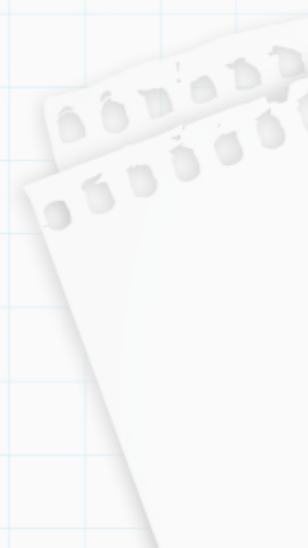
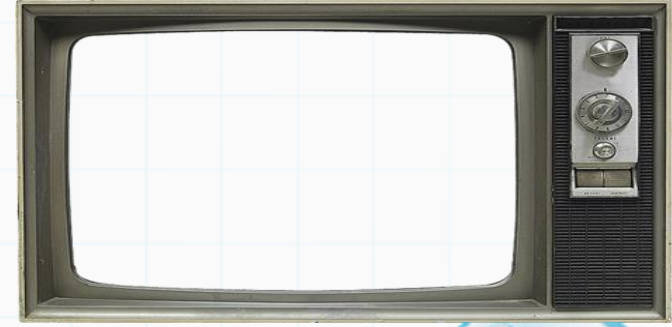
3	4	5	1
---	---	---	---

i $i+1$

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

i $i+1$

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

vamos repetir a varredura

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

vamos repetir a varredura

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

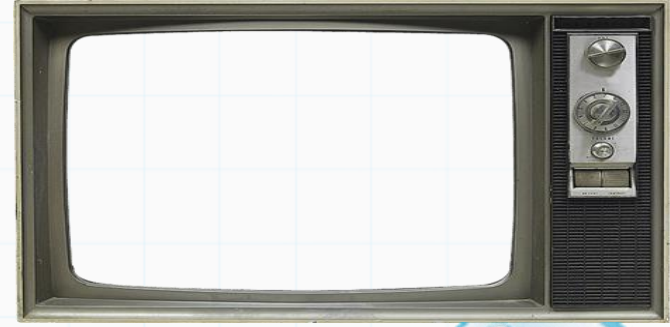
3	4	5	1
---	---	---	---

i $i+1$

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

i $i+1$

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

vamos repetir a varredura

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

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

quase lá



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

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

mais uma

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

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

mais uma

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

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

mais uma

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

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



Ordenação

Ordenação da Bolha: seja o seguinte vetor a ser ordenado de forma crescente

Vamos iterativamente varrer as posições adjacentes e toda vez que o par de elementos não estiverem ordenados vamos troca-los

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

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

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

mais uma

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

i $i+1$

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

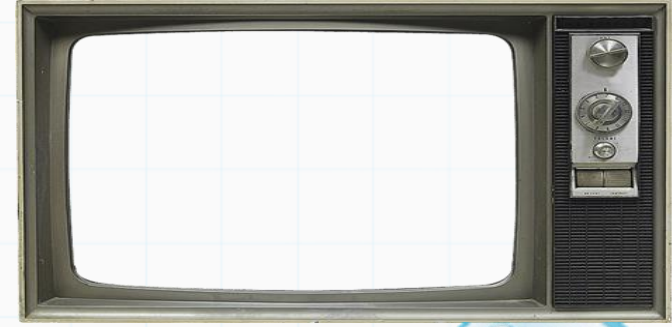
3	1	4	5
---	---	---	---

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

ordenado!

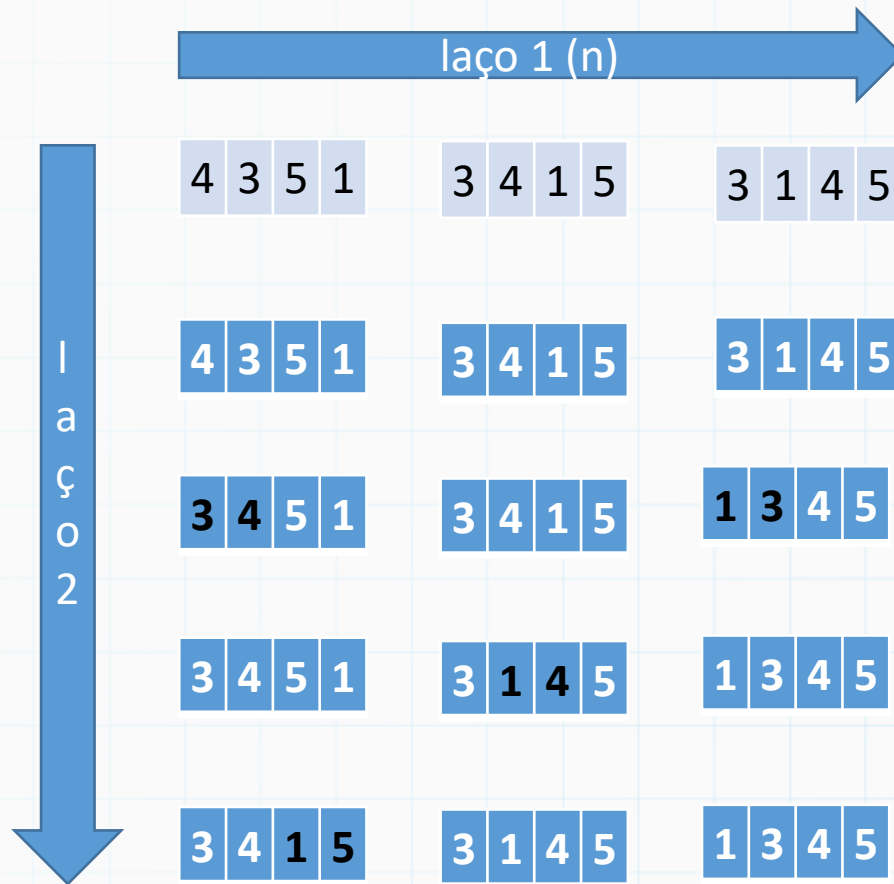
Teorema: É preciso no máximo n (tamanho do vetor) varreduras para ordenar um vetor

O nome do método vem devido aos elementos tenderem (flutuarem) a se mover para a posição correta como bolhas indo em direção da superfície.



Ordenação

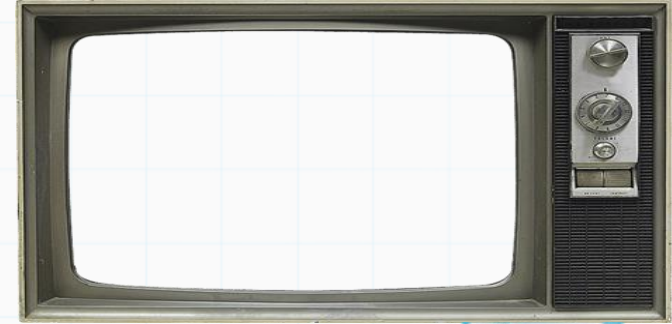
Ordenação da Bolha: O método pode ser implementado com laço duplo, um para as n varreduras e outro para realizar cada varredura. Vamos fazer ?



I percorre o vetor

J percorre o vetor

Varredura: troca as posições J e J+1 que não estiverem de acordo com a ordenação

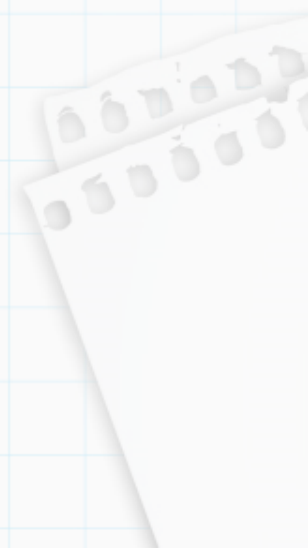
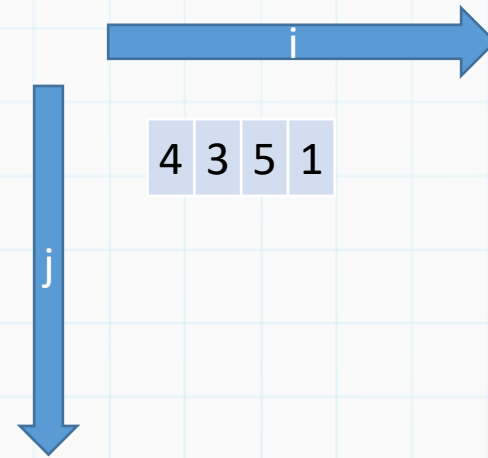


Ordenação

Ordenação da Bolha: O método pode ser implementado com laço duplo, um para as n varreduras e outro para realizar cada varredura. Vamos fazer ?

```
1 n = int(input("n?"))
2 v =[0]*(n)
3
4 for i in range(n):
5     v[i]=(int(input("num. "+str(i)+" ")))
6
7 for i in range(n):
8     for j in range(n-1):
9         if (v[j]>v[j+1]):
10             t      = v[j]
11             v[j]    = v[j+1]
12             v[j+1]  = t
13
14 print(v)
```

[código](#)



Ordenação

Ordenação da Bolha: O método pode ser implementado com laço duplo, um para as n varreduras e outro para realizar cada varredura. Vamos fazer ?

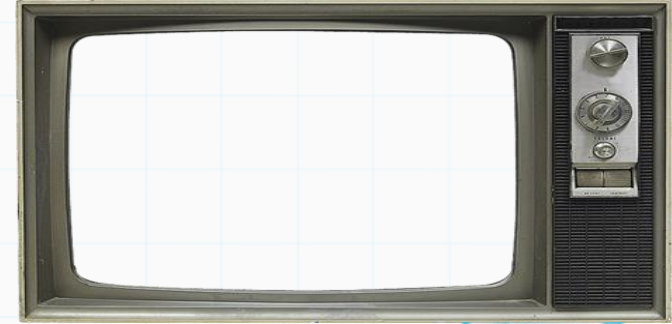
```
1 n = int(input("n?"))
2 v =[0]*(n)
3
4 for i in range(n):
5     v[i]=(int(input("num. "+str(i)+" ")))
6
7 for i in range(n):
8     for j in range(n-1):
9         if (v[j]>v[j+1]):
10             t      = v[j]
11             v[j]    = v[j+1]
12             v[j+1]  = t
13
14 print(v)
```

[código](#)



```
if (v[j]>v[j+1]):
    v[j]    = v[j+1]
    v[j+1]  = v[j]
```

..	4	3	..
..	3	3	..
..	3	3	..



Ordenação

Ordenação da Bolha: O método pode ser implementado com laço duplo, um para as n varreduras e outro para realizar cada varredura. Vamos fazer ?

```
1 n = int(input("n?"))
2 v =[0]*(n)
3
4 for i in range(n):
5     v[i]=(int(input("num. "+str(i)+" ")))
6
7 for i in range(n):
8     for j in range(n-1):
9         if (v[j]>v[j+1]):
10             t      = v[j]
11             v[j]    = v[j+1]
12             v[j+1]  = t
13
14 print(v)
```

[código](#)

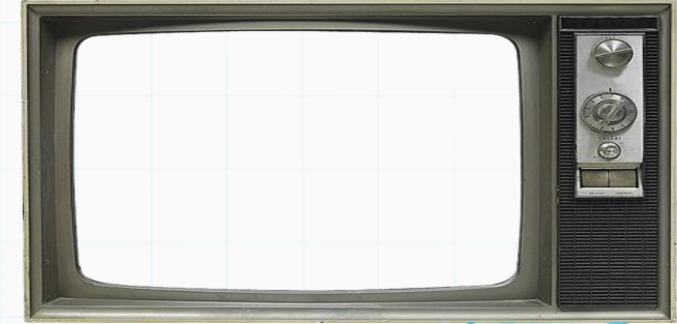


```
if (v[j]>v[j+1]):
    v[j]    = v[j+1]
    v[j+1]  = v[j]
```

..	4	3	..
..	3	3	..
..	3	3	..

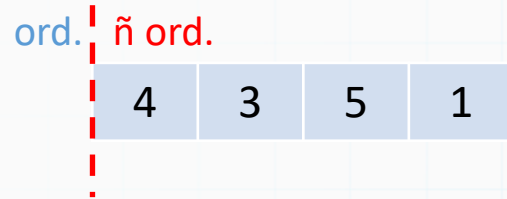
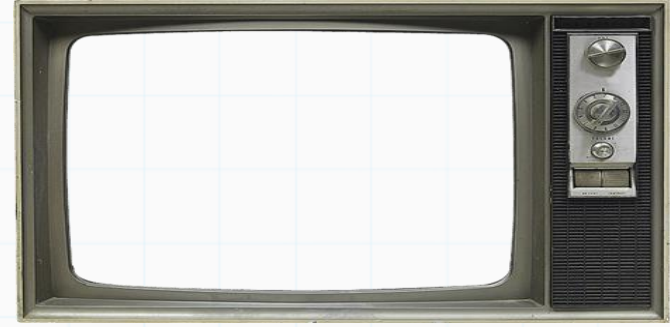


```
if (v[j]>v[j+1]):
    v[j],v[j+1] = v[j+1],v[j]
```



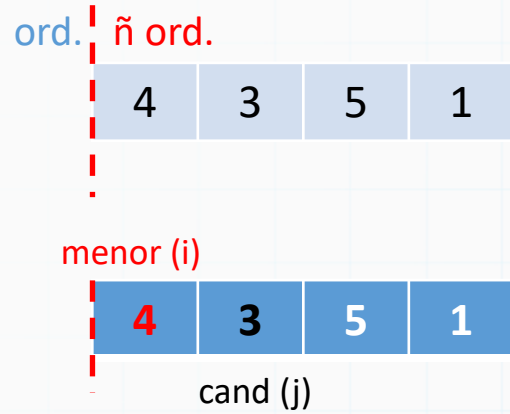
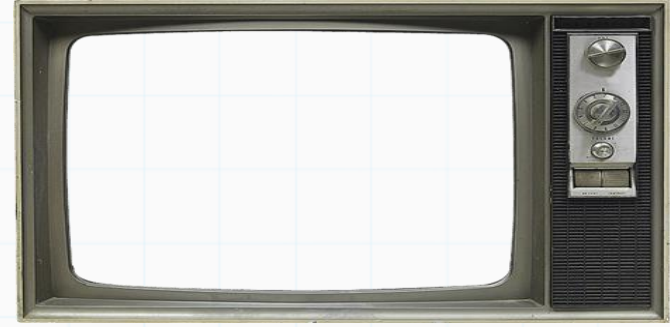
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



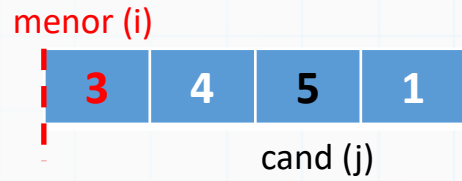
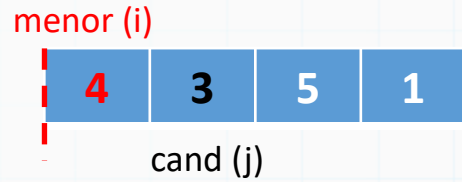
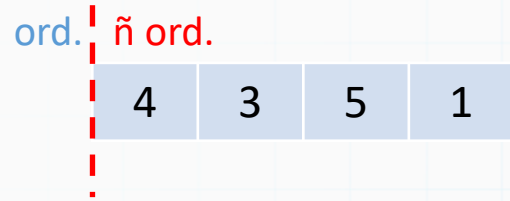
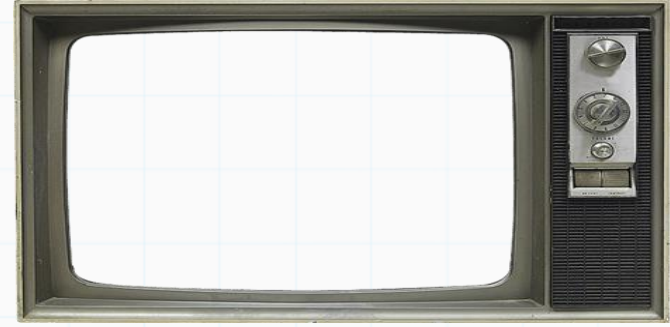
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



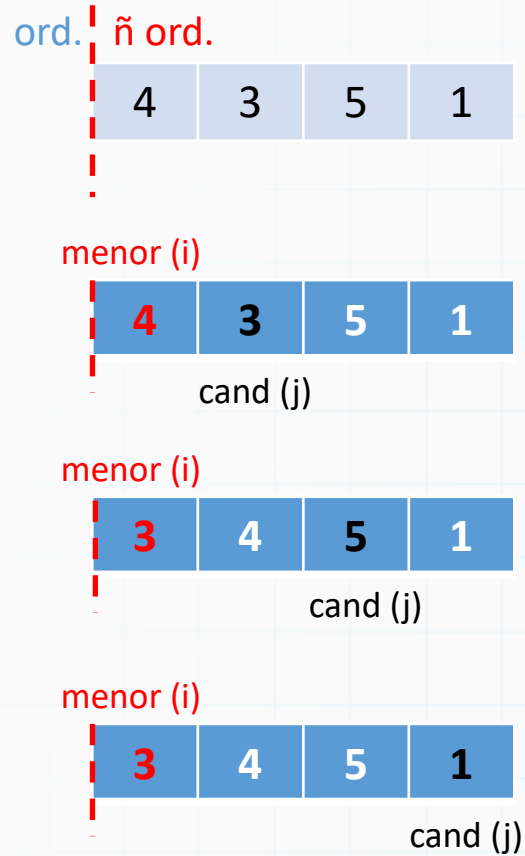
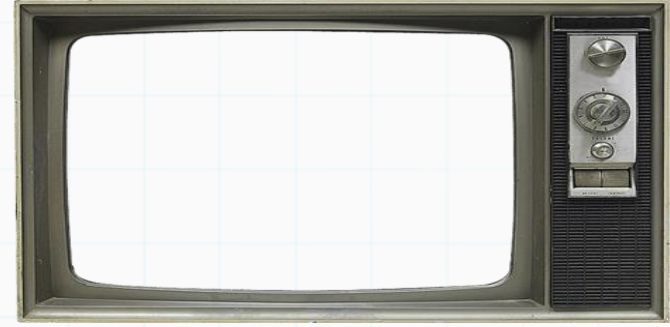
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



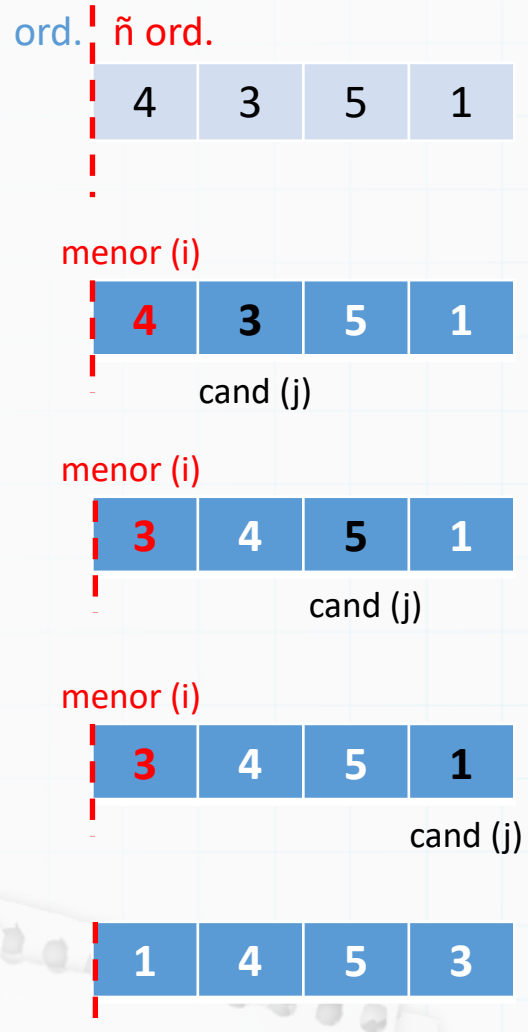
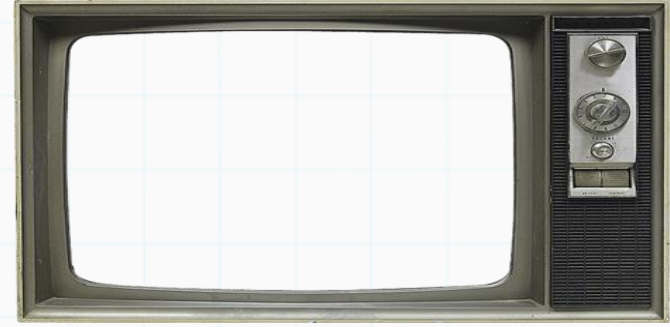
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



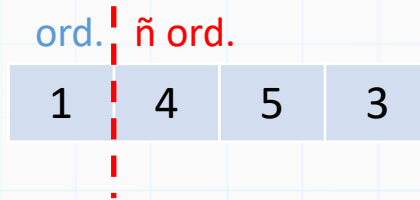
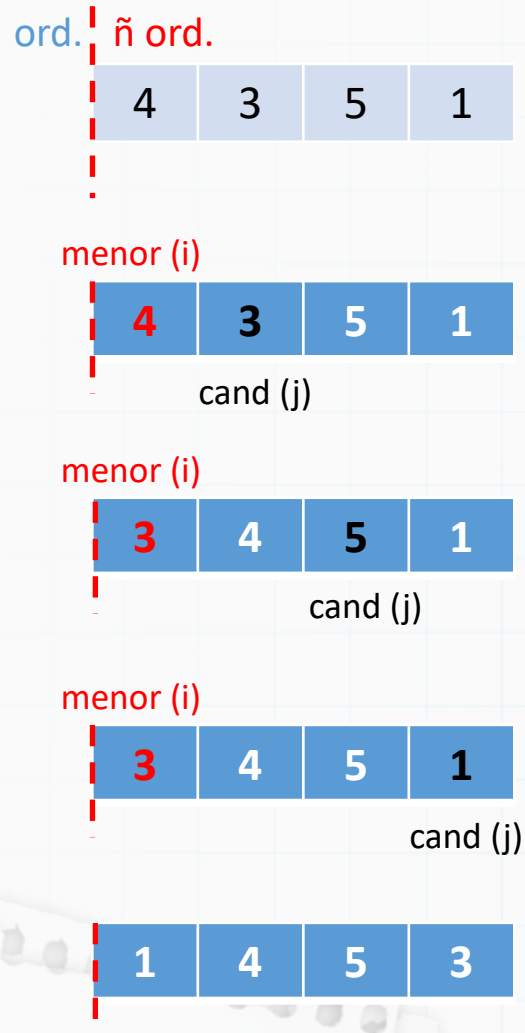
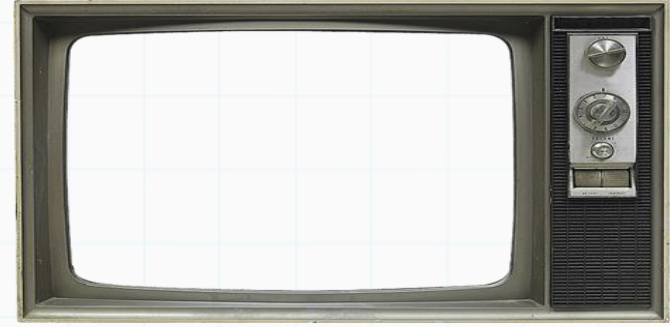
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



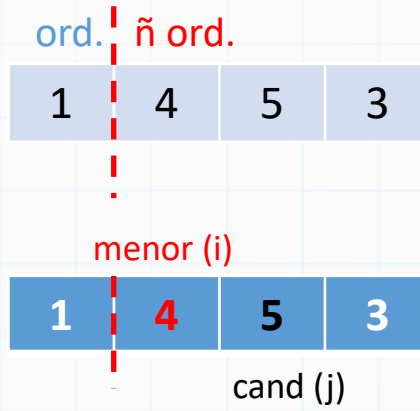
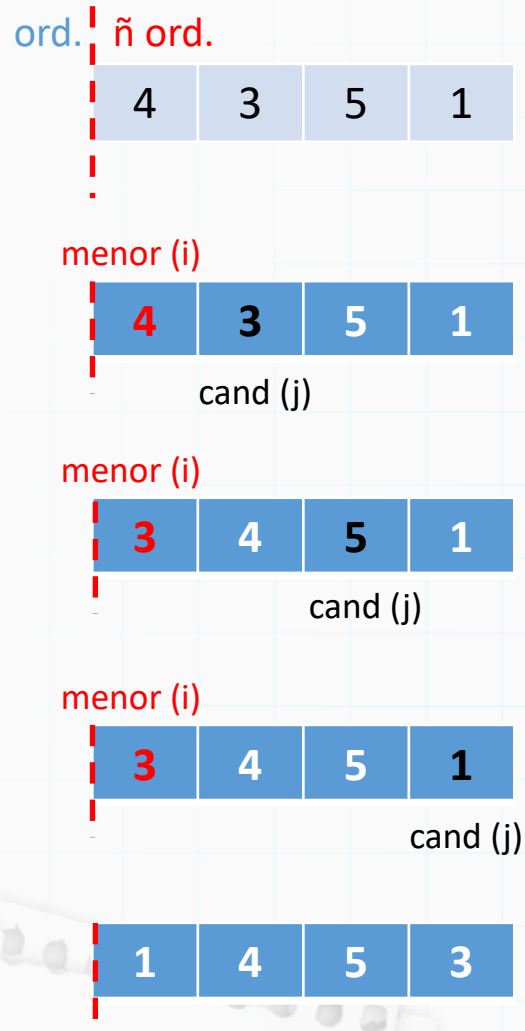
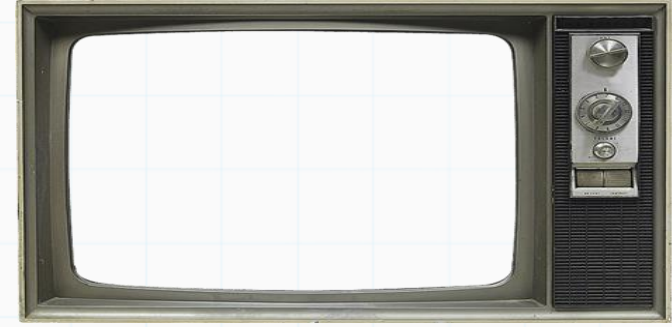
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



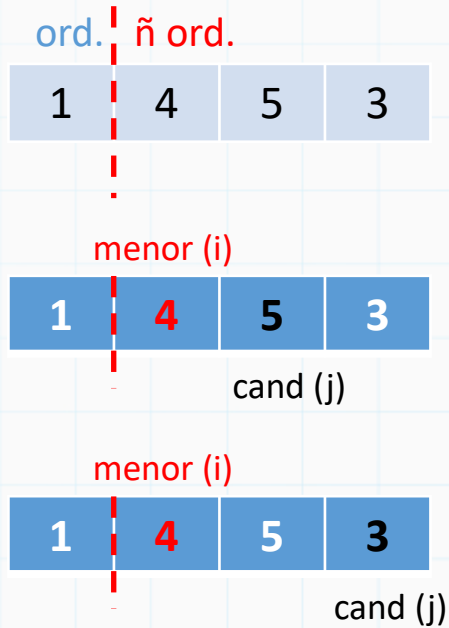
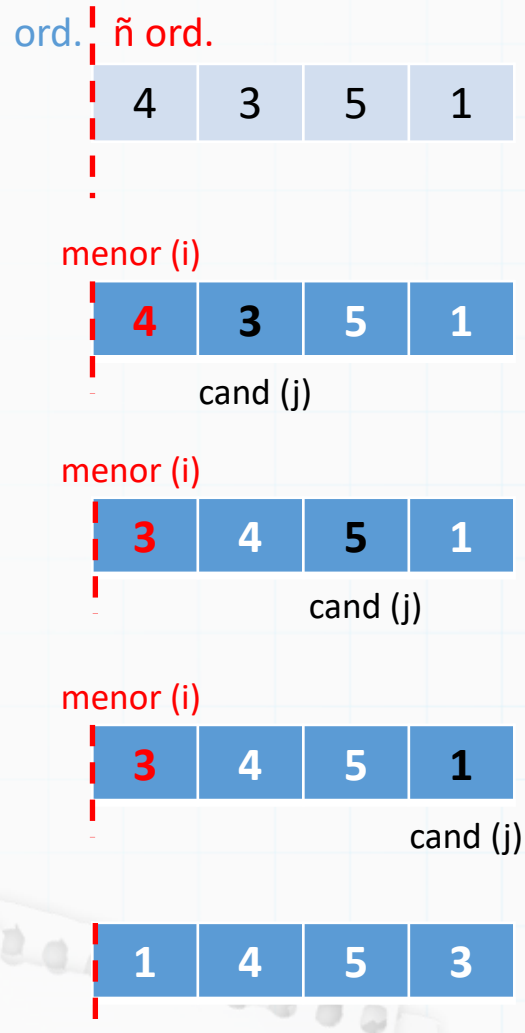
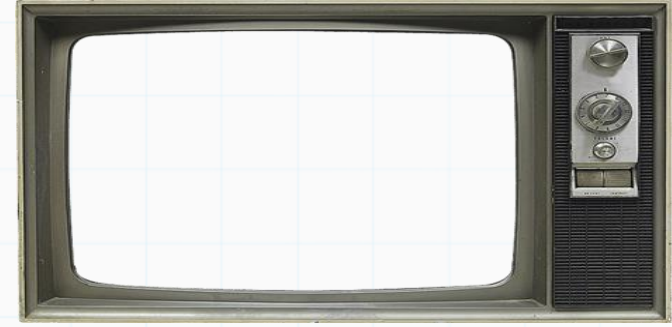
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



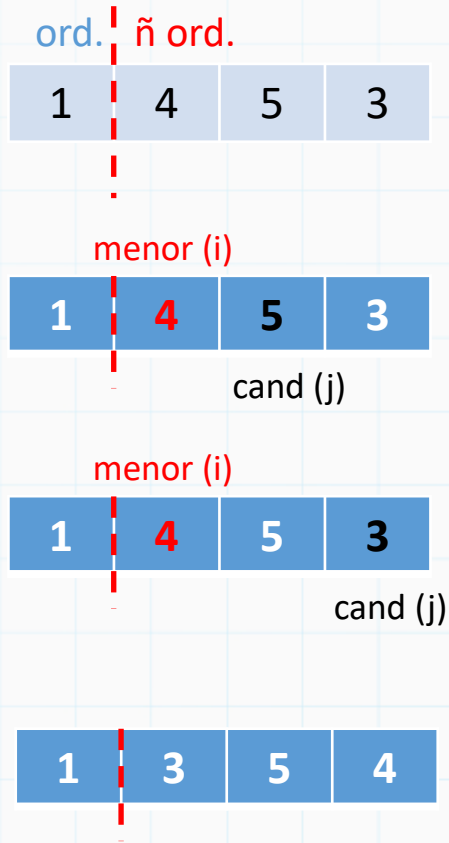
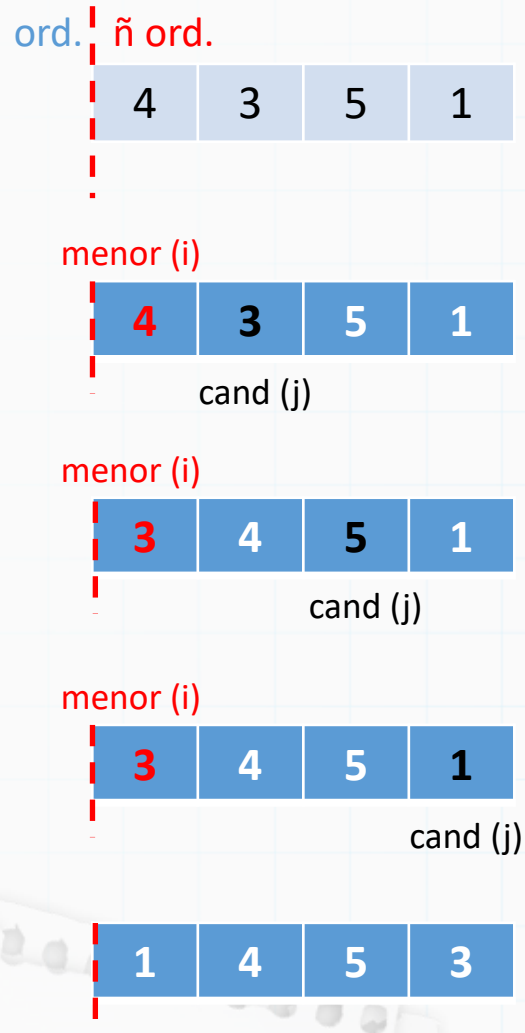
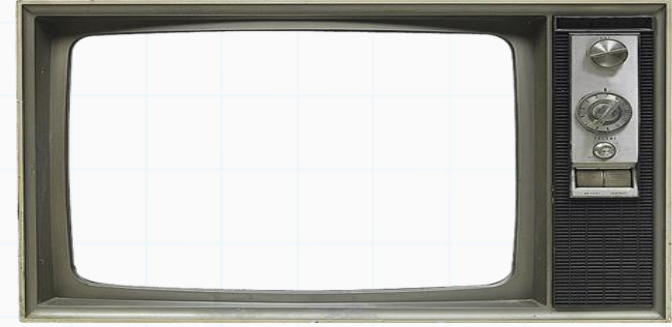
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



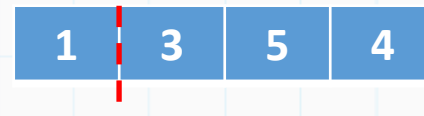
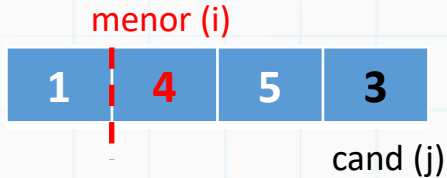
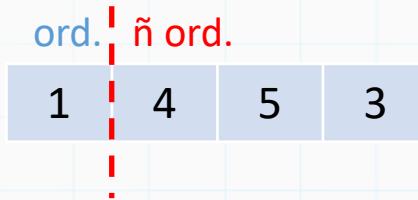
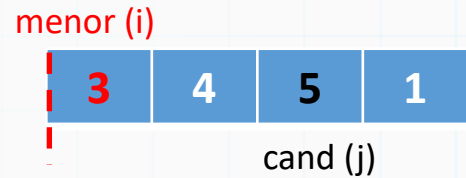
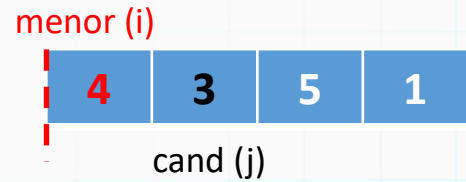
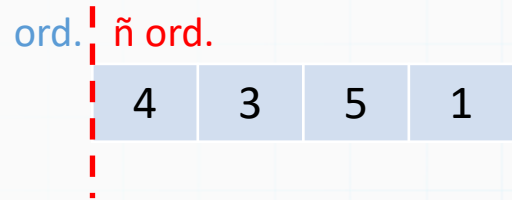
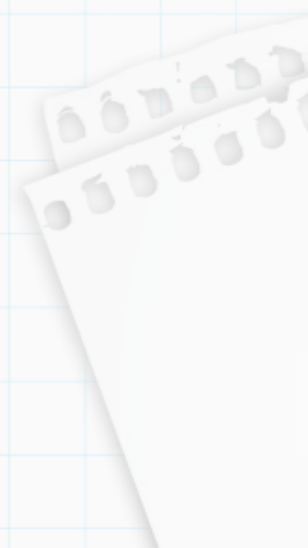
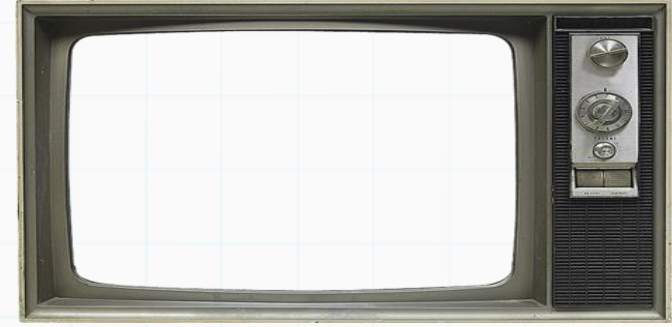
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



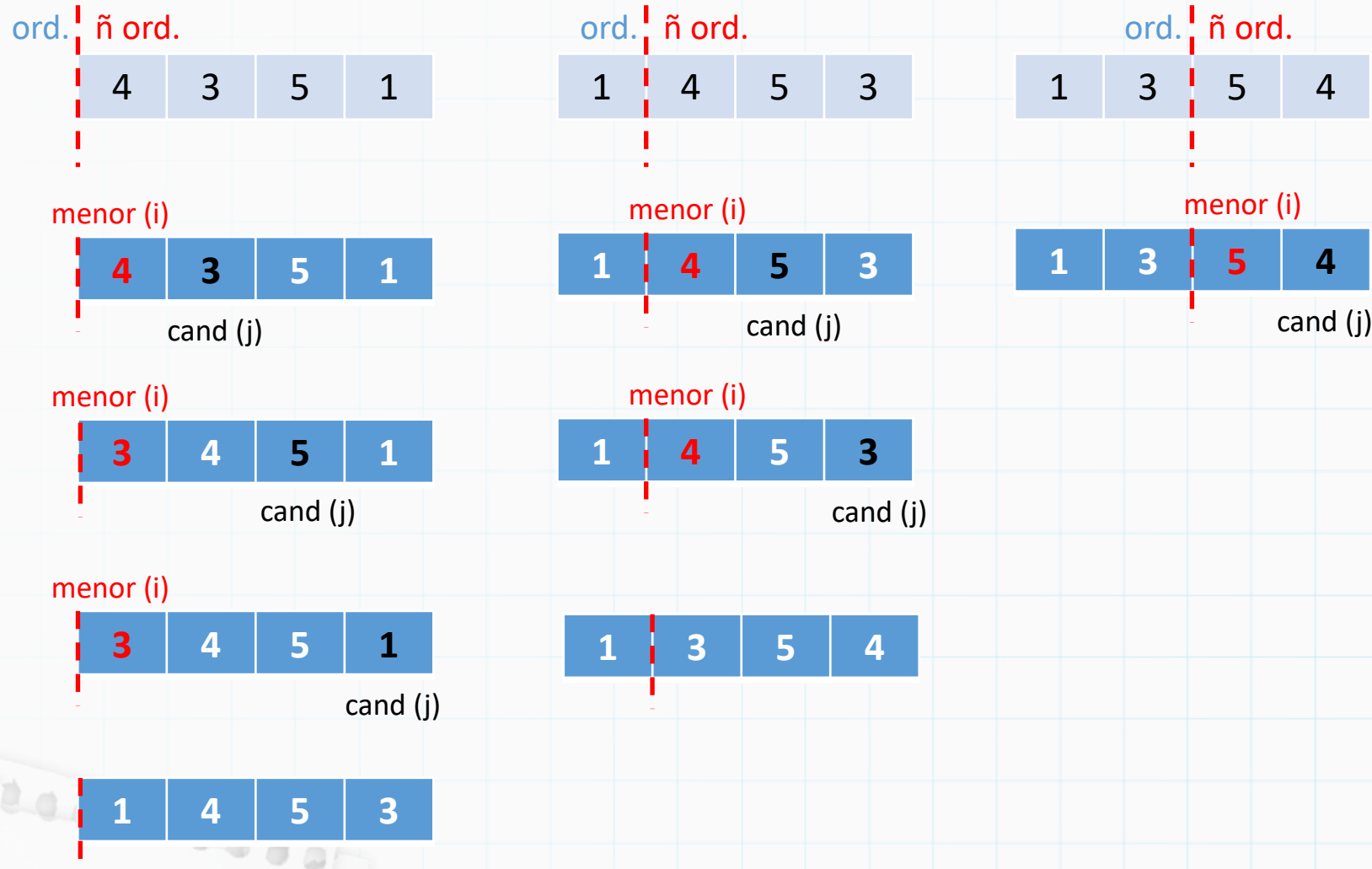
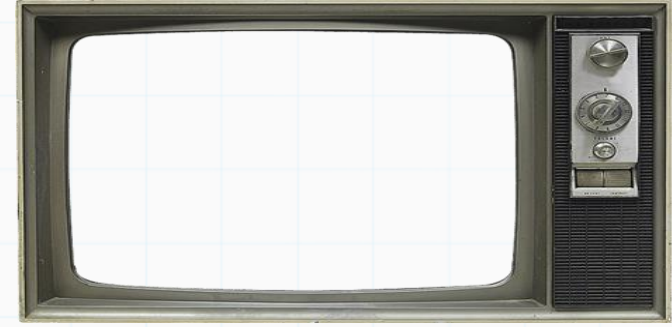
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



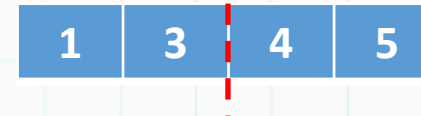
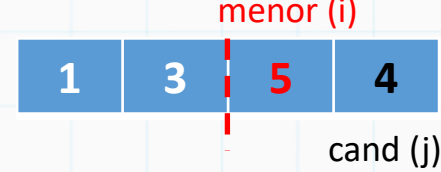
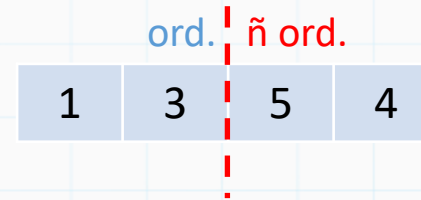
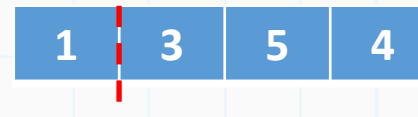
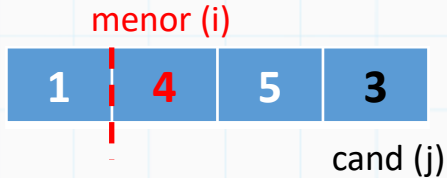
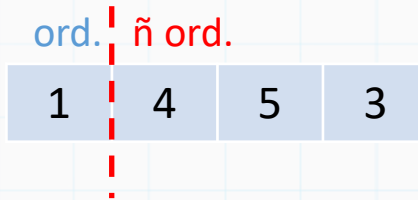
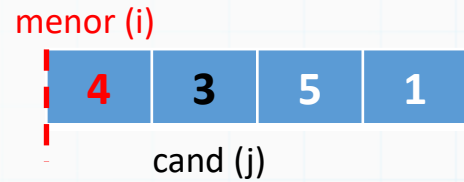
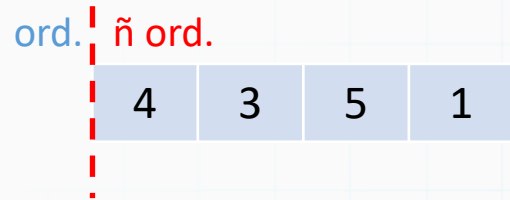
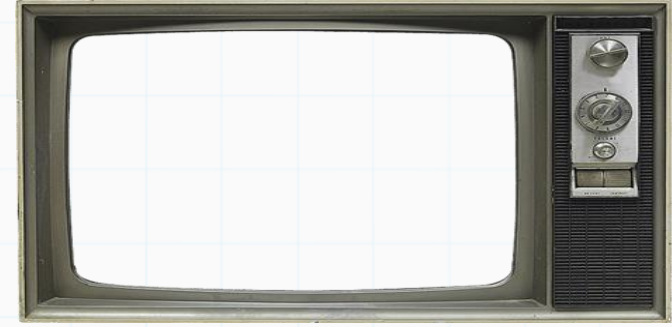
Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.

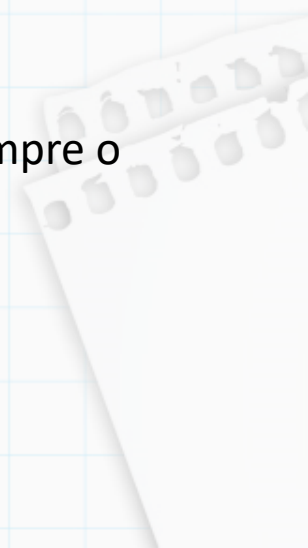


Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.

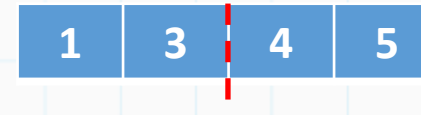
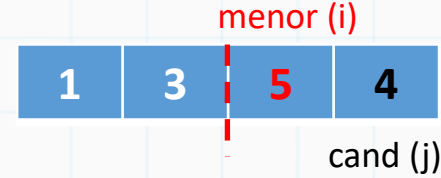
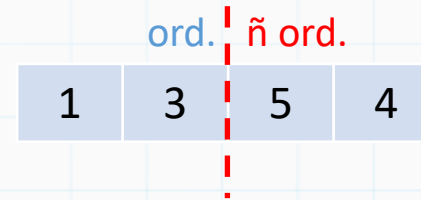
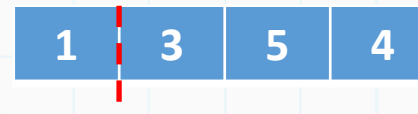
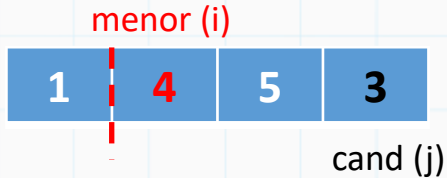
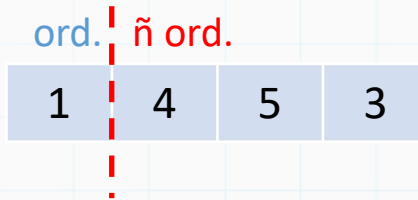
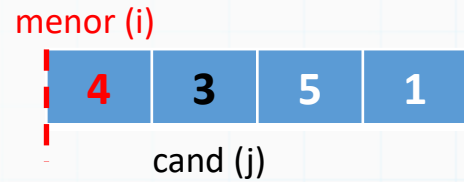
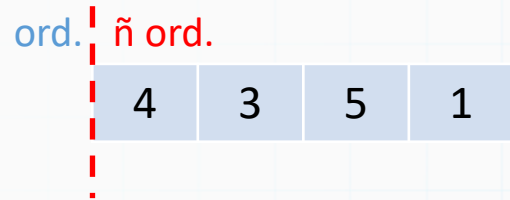
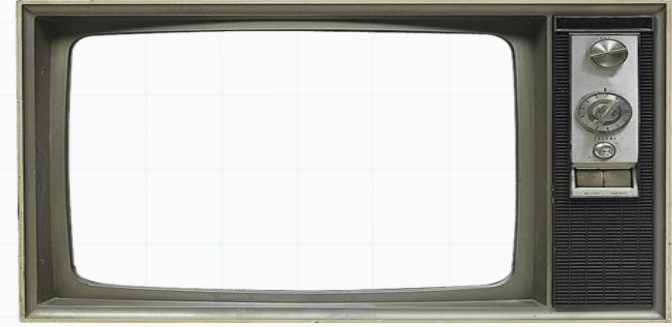


- Percorremos o vetor (n-1) vezes, procurando sempre o i-ésimo menor elemento



Ordenação

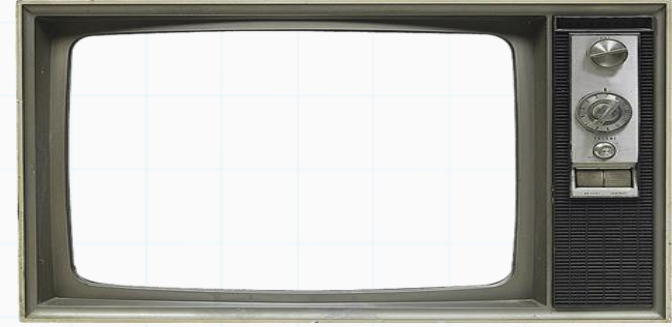
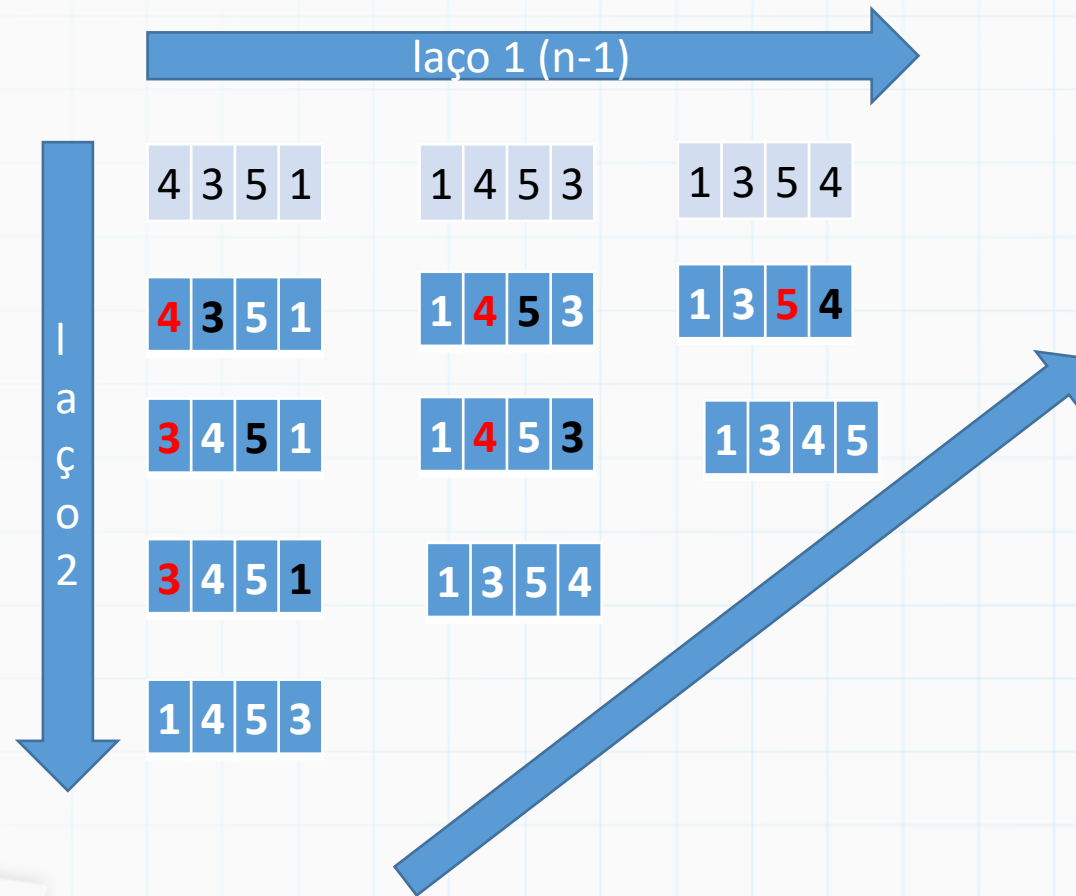
Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada.



- Percorremos o vetor (n-1) vezes, procurando sempre o i-ésimo menor elemento
- Repare que o número de comparações diminui a cada iteração

Ordenação

Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada. Vamos fazer ?



I percorre o vetor

J percorre o vetor a partir da posição I+1

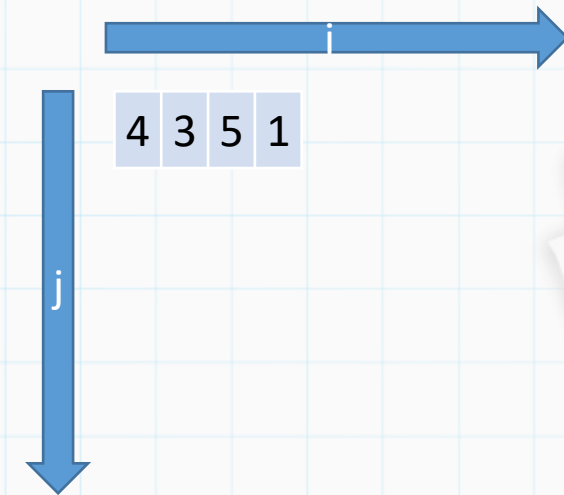
Varredura: troca as posições I e J que não estiverem de acordo com a ordenação

Ordenação

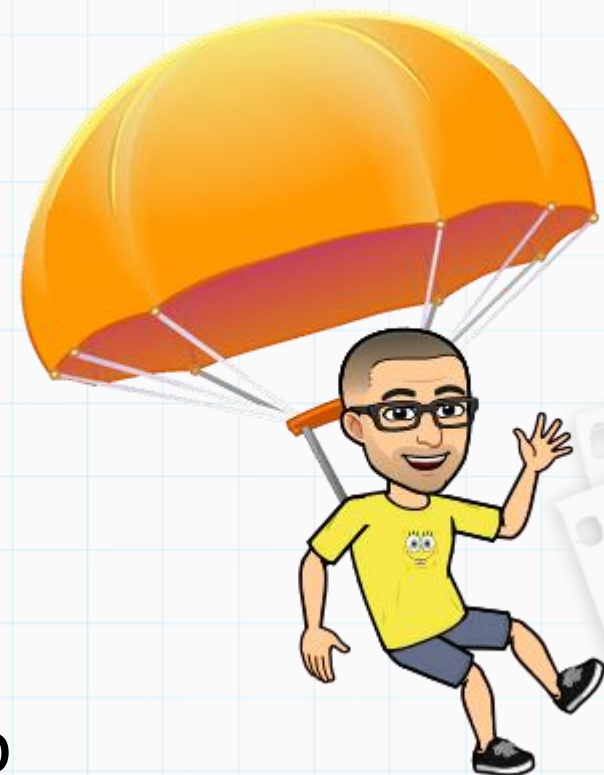
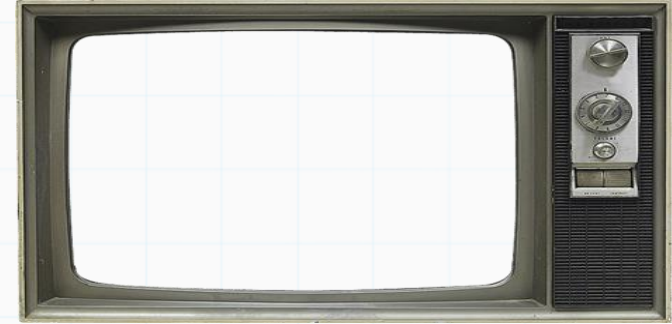
Ordenação por seleção: a lista está dividida em parte ordenada e não ordenada, a cada iteração, encontramos o menor elemento da parte não ordenada para colocar na ordenada. Vamos fazer ?

```
1 n = int(input("n?"))
2 v = [0]*(n)
3
4 for i in range(n):
5     v[i]=(int(input("num. "+str(i)+" ")))
6
7 for i in range(n-1):
8     for j in range(i+1,n):
9         if (v[i]>v[j]):
10             t = v[i]
11             v[i] = v[j]
12             v[j] = t
13 print(v)
```

[código](#)



Até a próxima



Slides baseados no curso de Vanessa Braganholo