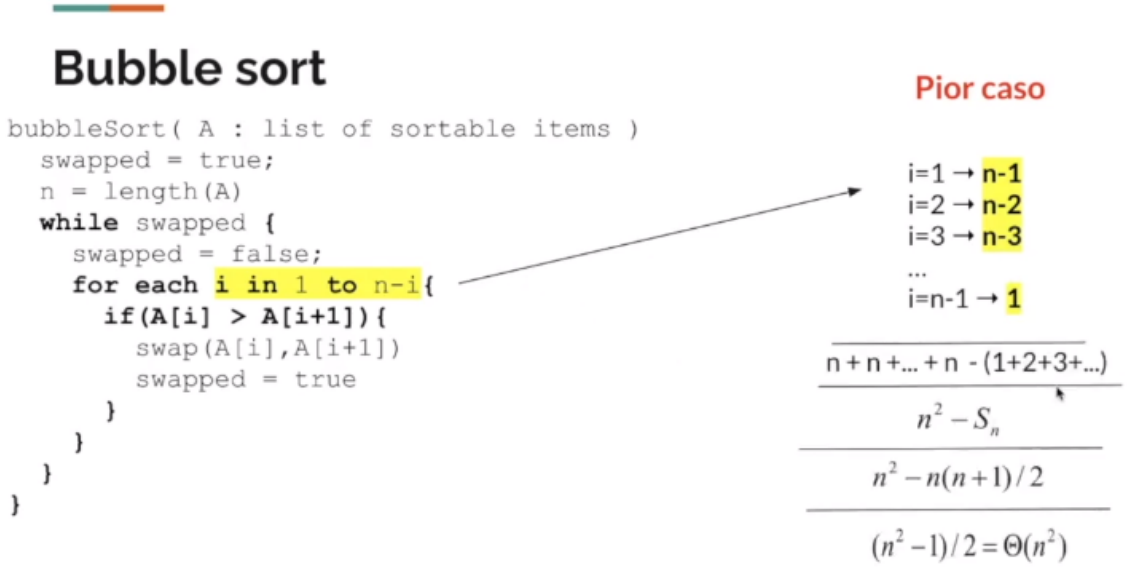
BubbleSort:

Baseado na **troca de elementos adjacentes**.

Cada iteração leva o elemento maior para o final do vetor.

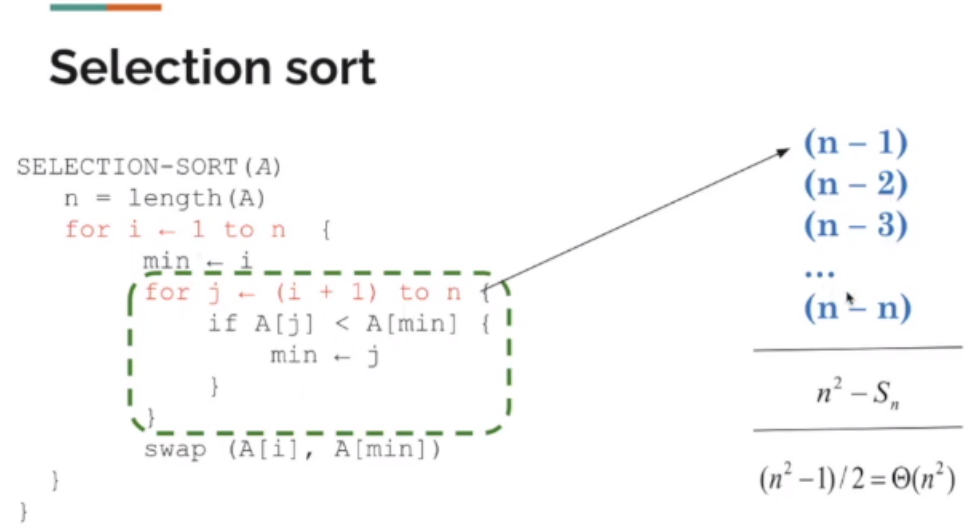


Melhor caso é na ordem de n; mas vai iterar na ordem de n vezes.

SelectionSort:

Baseado na seleção do maior e seu posicionamento no final.

Primeiro seleciona, depois leva para a posição correta.



InsertionSort:

Baseado na ideia de inserir um elemento em uma lista ordenada.

Insertion sort

```
INSERTION-SORT(A, n)
    for j ← 2 to n do
        key ← A[j]
        i ← j-1
        while i > 0 and A[i] > key do
            A[i+1] ← A[i]
            i ← i-1
        A[i+1] = key
```

(1)
(2)
(3)
...
(n)

Comparação

Algoritmo	melhor tempo	tempo médio	pior tempo	espaço
Insertion sort	O(n)	O(n^2)	O(n^2)	in-place
Selection sort	O(n^2)	O(n^2)	O(n^2)	in-place
Bubble sort	O(n)	O(n^2)	O(n^2)	in-place

BubbleSort muitas comparação e troca.
SelectionSort muitas comparações e menos trocas.
InsertionSort menos comparações (último da lista ordenada) e reduz a quantidade de troca (faz trocas suficientes para encaixar a posição do elemento na lista ordenada).

Questões de Implementação:

Como ordenar apenas uma faixa de um array ?