

Paso 1:

Create(A)  
Sort

N = 5

Random

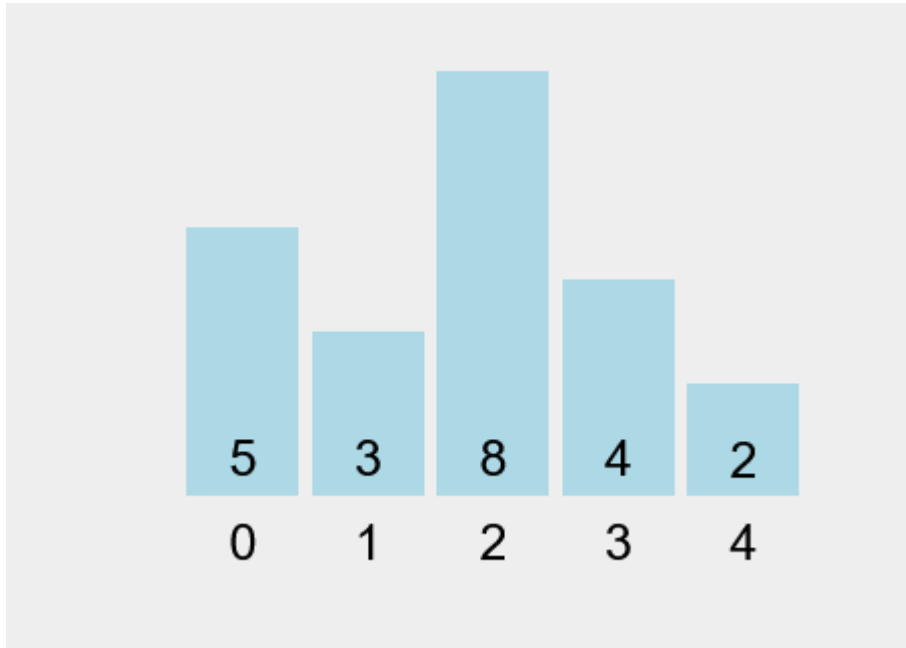
Sorted

Nearly sorted

Many Duplicates

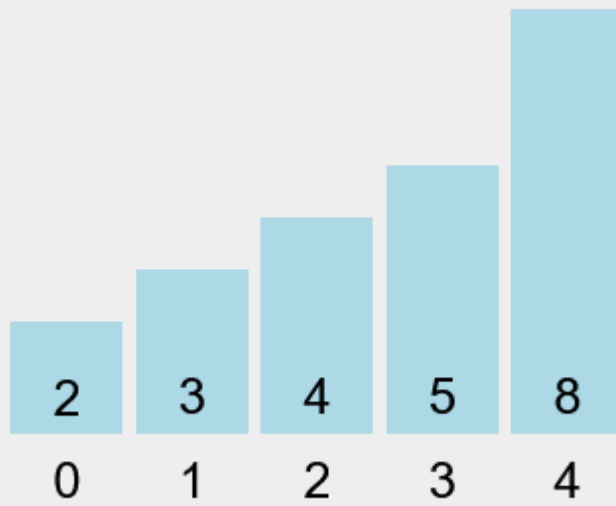
A = 5,3,8,4,2

Go



Paso 2:

Sort



List is sorted!  
Inversion Index = 7.



```
do
  swapped = false
  for i = 1 to indexOfLastUnsortedElement-1
    if leftElement > rightElement
      swap(leftElement, rightElement)
      swapped = true; ++swapCounter
  while swapped
```



### Preguntas de reflexión

**¿Cuántos intercambios se realizaron?**

R = 7

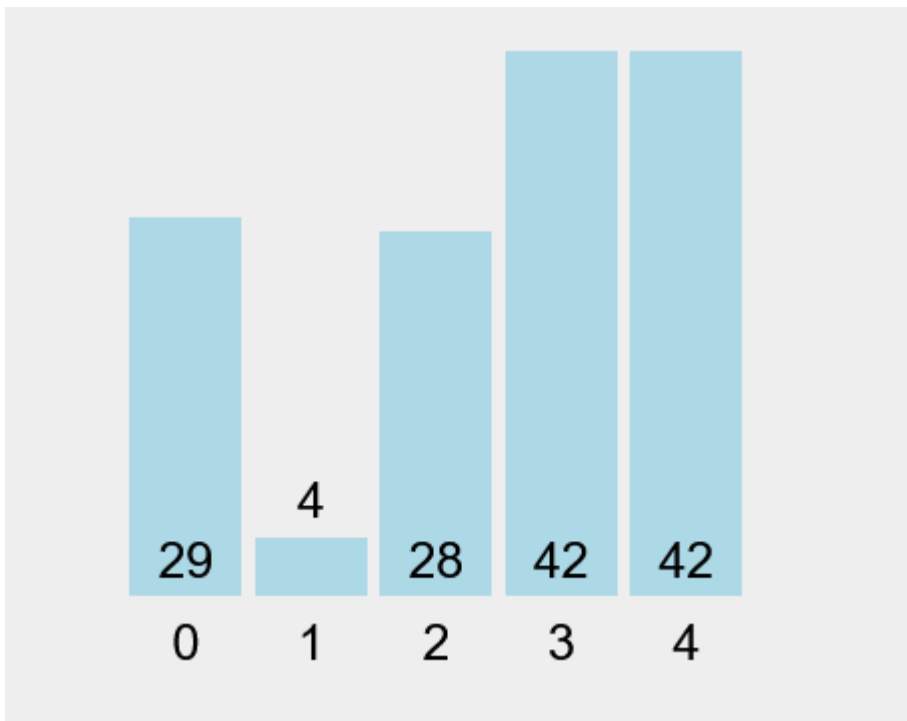
**¿Qué pasa si la lista ya está ordenada?**

R = No se realiza ningún intercambio.

**¿Qué eficiencia tiene este algoritmo frente a listas grandes?**

R =  $O(n^2)$

### Desafío 1:

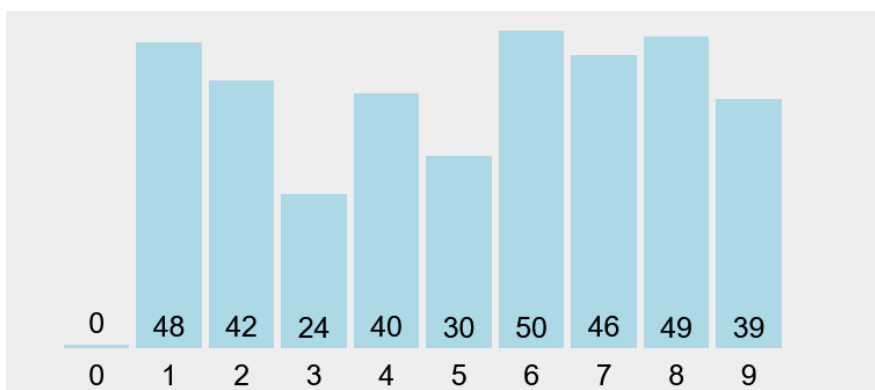


### Elementos que se intercambiaron:

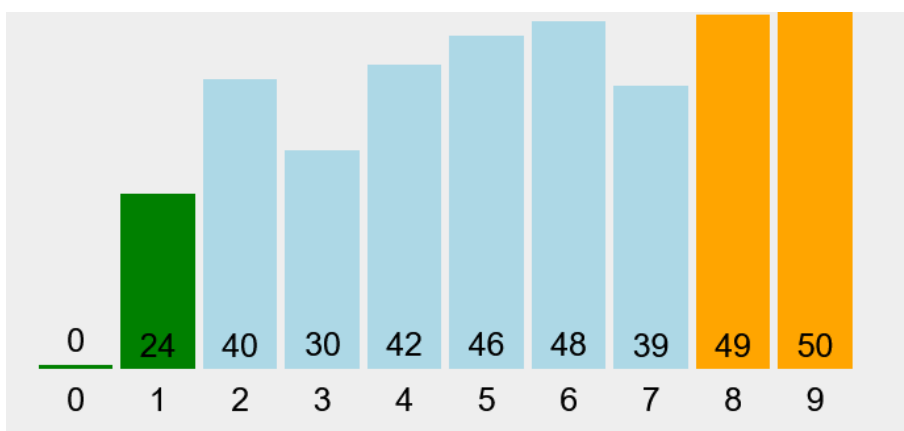
1. Se comparó 29 y 4, se realizó intercambio
2. Se comparó 29 con 28, se realizó intercambio
3. Se comparó 29 con 42, no se realizó intercambio
4. Se comparó 42 con 42, no se realizó intercambio
5. Se realiza una última comparación entre todos los elementos de la lista

## Desafío 2:

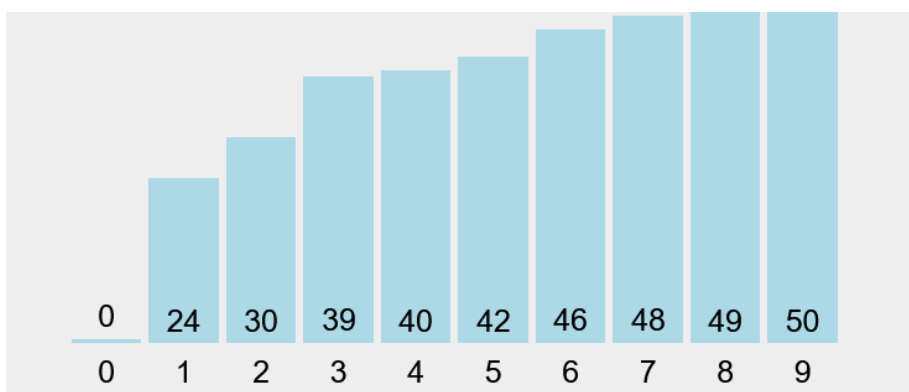
### Estado inicial:



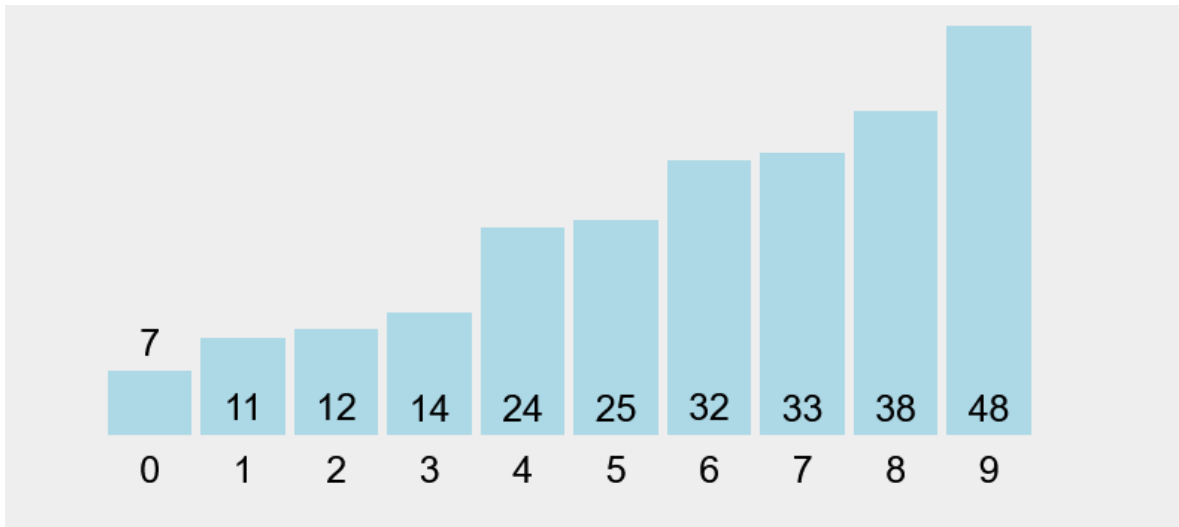
### Estado después de la mitad del proceso:



### Estado final:



### Desafío 3:



#### 9 comparaciones 0 intercambios

Algoritmo:

Comparar elementos en la posición 0 y 1.

Si elemento en la posición 0 es mayor al elemento en posición 1 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 1 y 2.

Si elemento en la posición 1 es mayor al elemento en posición 2 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 2 y 3.

Si elemento en la posición 2 es mayor al elemento en posición 3 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 3 y 4.

Si elemento en la posición 3 es mayor al elemento en posición 4 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 4 y 5.

Si elemento en la posición 4 es mayor al elemento en posición 5 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 5 y 6.

Si elemento en la posición 5 es mayor al elemento en posición 6 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 6 y 7.

Si elemento en la posición 6 es mayor al elemento en posición 7 se realiza intercambio. Si no se mantienen igual.

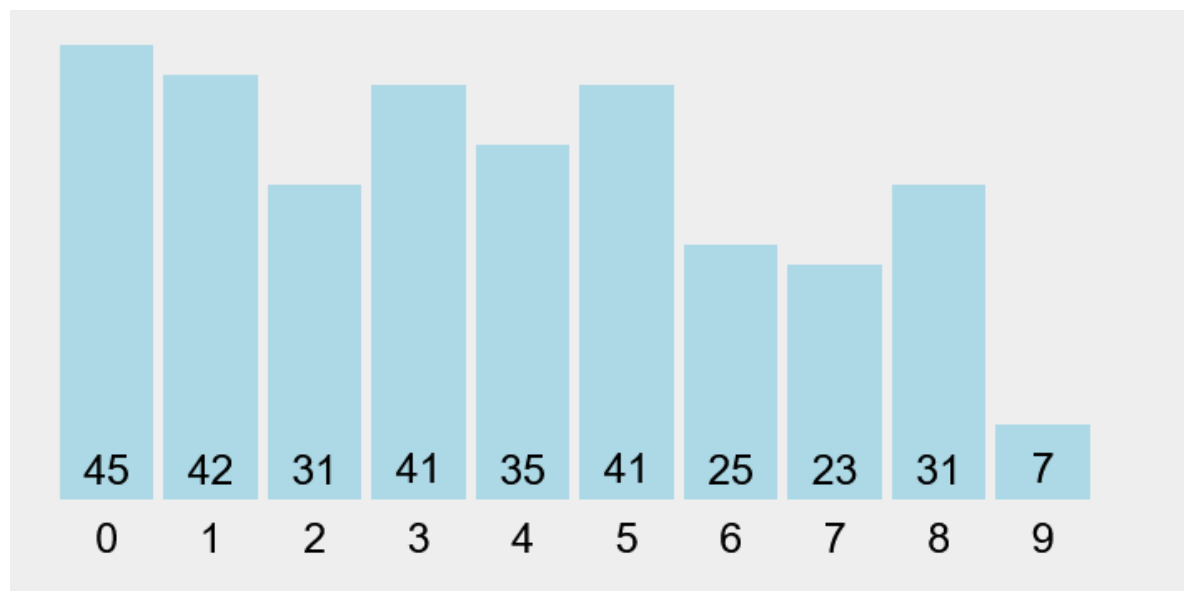
Comparar elementos en la posición 7 y 8.

Si elemento en la posición 7 es mayor al elemento en posición 8 se realiza intercambio. Si no se mantienen igual.

Comparar elementos en la posición 8 y 9.

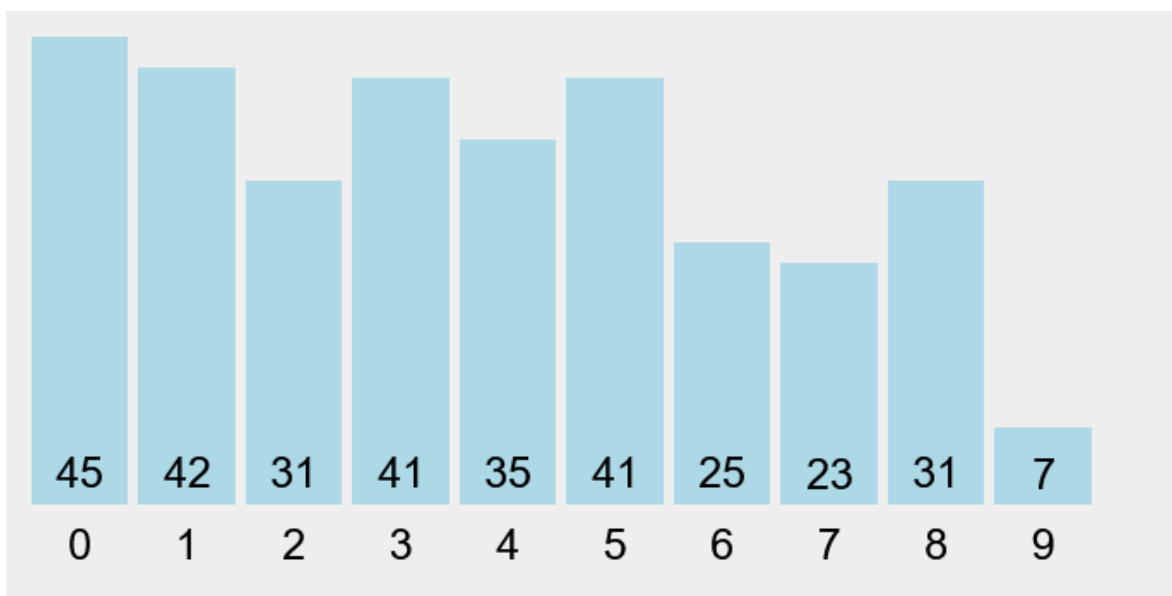
Si elemento en la posición 8 es mayor al elemento en posición 9 se realiza intercambio. Si no se mantienen igual.

#### Desafío 4



Ordenamiento es = a  $O(n^2)$  en este caso seria  $O(10^2)$  que es igual a 100 ordenamientos pero solamente ocupo 32 intercambios y 42 comparaciones.

**Desafío 5:**



**Con Bubble Sort 42 comparaciones y 32 intercambios.**

**Por Inserción 42 comparaciones 9 intercambios ya que el elemento es comparado con todos los elementos hasta encontrar la posición adecuada para ella.**

