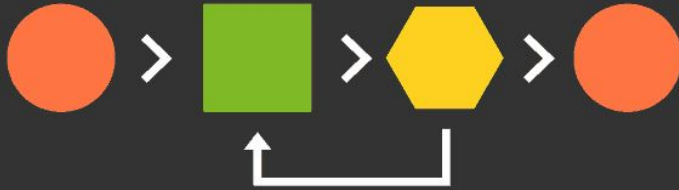


ALGORITMOS DE ORDENAMIENTO



- ❖ Insert Sort
- ❖ Bucket Sort

- Nicolás Figueroa Gómez
- David Iazo Pampa

Insertion Sort

Array

/

Vector

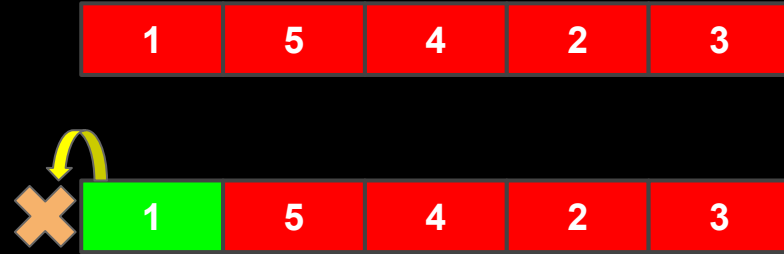
1	5	4	2	3
---	---	---	---	---

Insertion Sort

Array

/

Vector

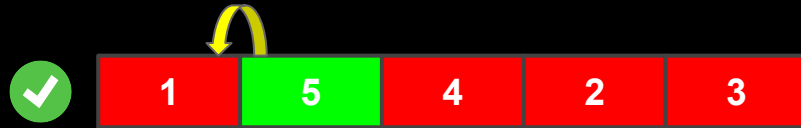


Insertion Sort

Array

/

Vector

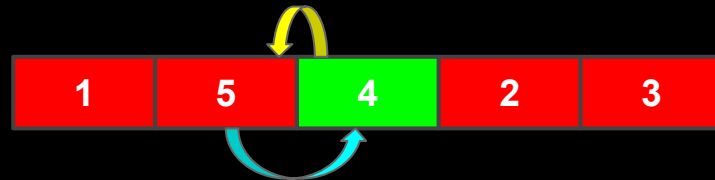
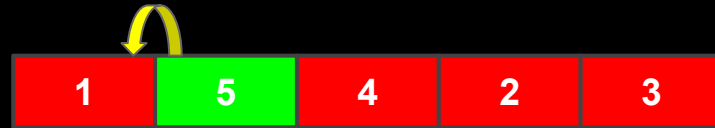


Insertion Sort

Array

/

Vector

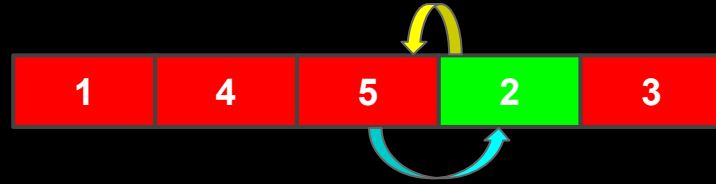


Insertion Sort

Array

/

Vector

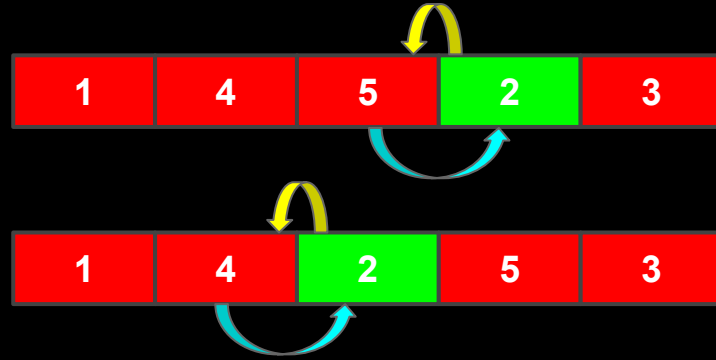


Insertion Sort

Array

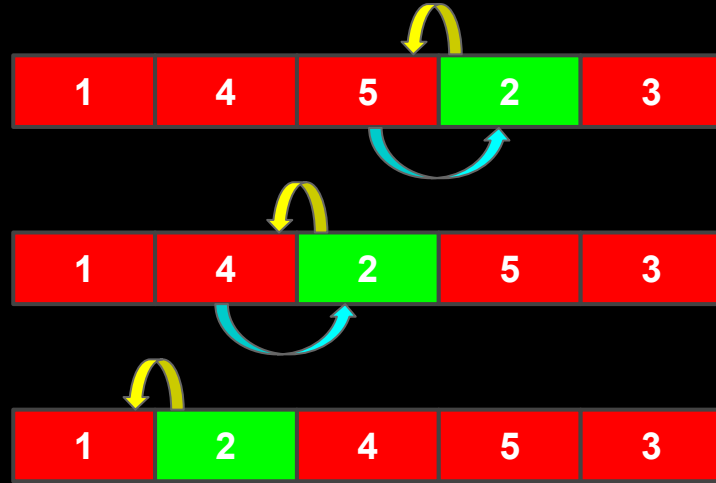
/

Vector



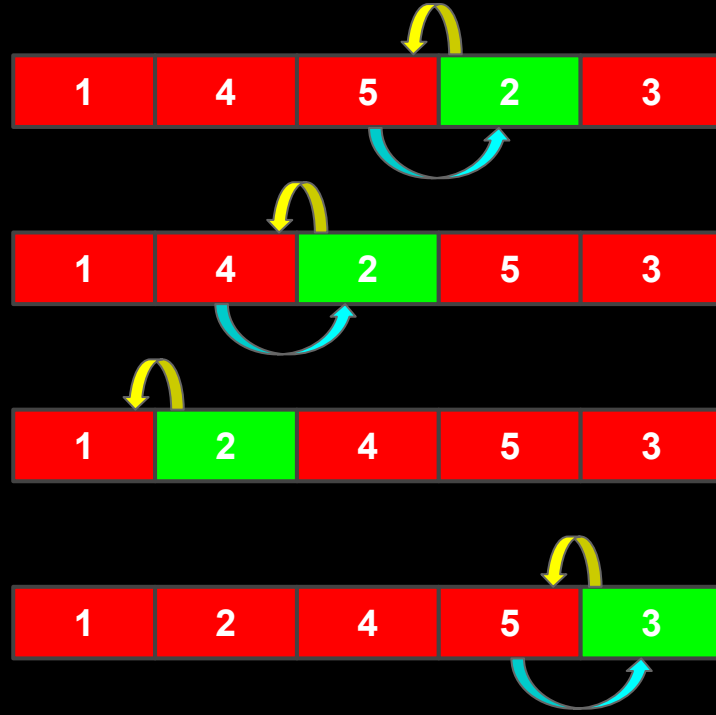
Insertion Sort

Array
/
Vector



Insertion Sort

Array
/
Vector

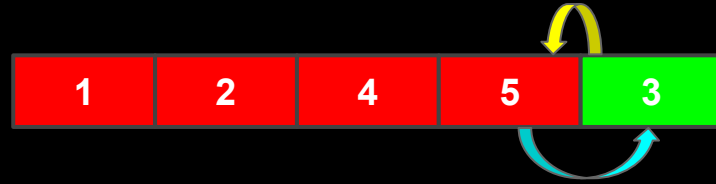


Insertion Sort

Array

/

Vector

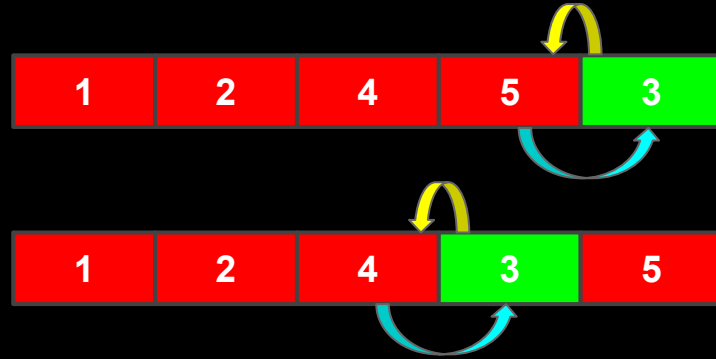


Insertion Sort

Array

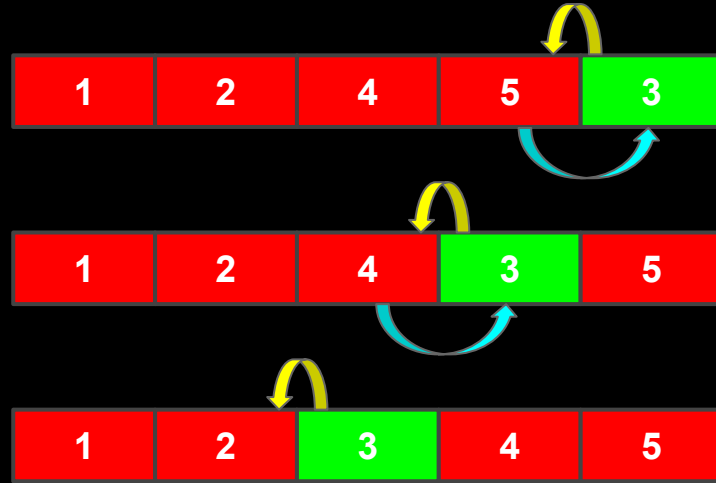
/

Vector



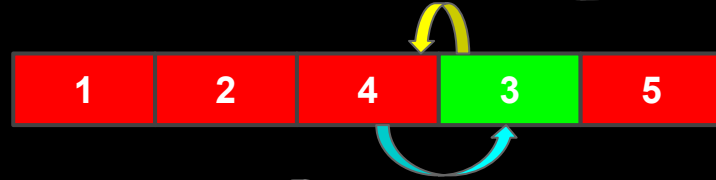
Insertion Sort

Array
/
Vector



Insertion Sort

Array
/
Vector



Final ➡



Limitations

- ▶ **Mejor:**

- ▶ Cuando los elementos ya están ordenados.

- ▶ **Peor:**

- ▶ Cuando los elementos se encuentran en orden inverso.
- ▶ Deficientes en grandes listas.
- ▶ Numerosas comparaciones.

Complexity and Stability

- ▶ Best case:
 - ▷ $O(n)$
- ▶ Space Complexity:
 - ▷ $O(n^2)$
- ▶ Worst case:
 - ▷ $O(n^2)$
- ▶ Stability:
 - ▷ Yes

Show me the code



Bucket Sort

Consideraremos el siguiente
array

11	75	2	81	50	32	45
----	----	---	----	----	----	----

Su calificación será de menor a
mayor

Bucket Sort

Array [6]
Elementos

0	1	2	3	4	5	6
11	75	2	81	50	32	45

Número total de elementos 7 -> $N=7$
Valor Max = 81 & Min = 2

Bucket Sort

Copy

0	copy_array [6]
1	copy_array [6]
2	copy_array [6]
3	copy_array [6]
4	copy_array [6]
5	copy_array [6]
6	copy_array [6]
7	copy_array [6]
8	copy_array [6]
9	copy_array [6]

Para ordenar los elementos
tomaremos

10Bucket-cubos 0-9

*Como es copi la cantidad
de elementos es del
original

Bucket Sort

Copy

0
1
2
3
4
5
6
7
8
9

Array [6]
Elementos

N=7

Max = 81

Min = 2

bucket = 10

0	1	2	3	4	5	6
11	75	2	81	50	32	45

*Ahora el Divisor(Valor techo)
 $\text{divisor} = \text{ceil}((\text{max}+1)/\text{bucket})$
 $\text{divisor} = \text{ceil}((81+1)/10)$
 $\text{divisor} = \text{ceil}(8.2)$
 $\text{divisor} = 9$

Bucket Sort

Copy

0
1
2
3
4
5
6
7
8
9

Array [6]
Elementos

0	1	2	3	4	5	6
11	75	2	81	50	32	45

floor (valor piso)
 $\text{Copy}[j] = \text{Array}[i]$
 $j = \text{floor}(\text{Array}[i] / \text{divisor})$

$N=7$
Max = 81
Min = 2
bucket = 10
divisor = 9

Bucket Sort

Copy

0
1
2
3
4
5
6
7
8
9

Array [6]
Elementos

0	1	2	3	4	5	6
11	75	2	81	50	32	45

Array[0]

$j = \text{floor}(\text{Array}[i] / \text{divisor})$

$j = \text{floor}(11/9)$

$j = \text{floor}(1.2222)$

$j = 1$

Copy[j]=11

Copy[1]=11

$N=7$

Max = 81

Min = 2

bucket = 10

divisor= 9

Bucket Sort

Copy

0
1
2
3
4
5
6
7
8
9



11

Array [6]
Elementos

0	1	2	3	4	5	6
11	75	2	81	50	32	45

Array[0]

$j = \text{floor}(\text{Array}[i] / \text{divisor})$

$j = \text{floor}(11/9)$

$j = \text{floor}(1.2222)$

$j = 1$

Copy[j]=11

Copy[1]=11

N=7

Max = 81

Min = 2

bucket = 10

divisor= 9

Bucket Sort

Copy

0
1
2
3
4
5
6
7
8
9



11

Array [6]
Elementos

0	1	2	3	4	5	6
11	75	2	81	50	32	45

$j = \text{floor}(11/9) \rightarrow j=1$

$j = \text{floor}(75/9) \rightarrow j=8$

$j = \text{floor}(2/9) \rightarrow j=0$

$j = \text{floor}(81/9) \rightarrow j=9$

$j = \text{floor}(50/9) \rightarrow j=5$

$j = \text{floor}(32/9) \rightarrow j=3$

$j = \text{floor}(45/9) \rightarrow j=5$

$N=7$

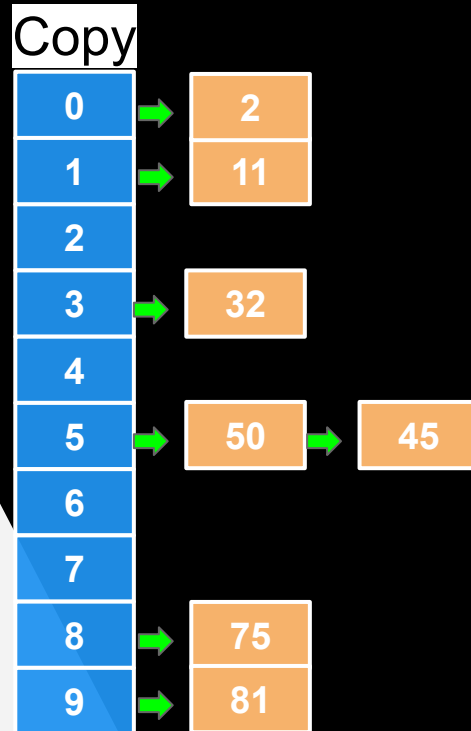
Max = 81

Min = 2

bucket = 10

divisor= 9

Bucket Sort



$j = \text{floor}(11/9) \rightarrow j = 1$

$j = \text{floor}(75/9) \rightarrow j = 8$

$j = \text{floor}(2/9) \rightarrow j = 0$

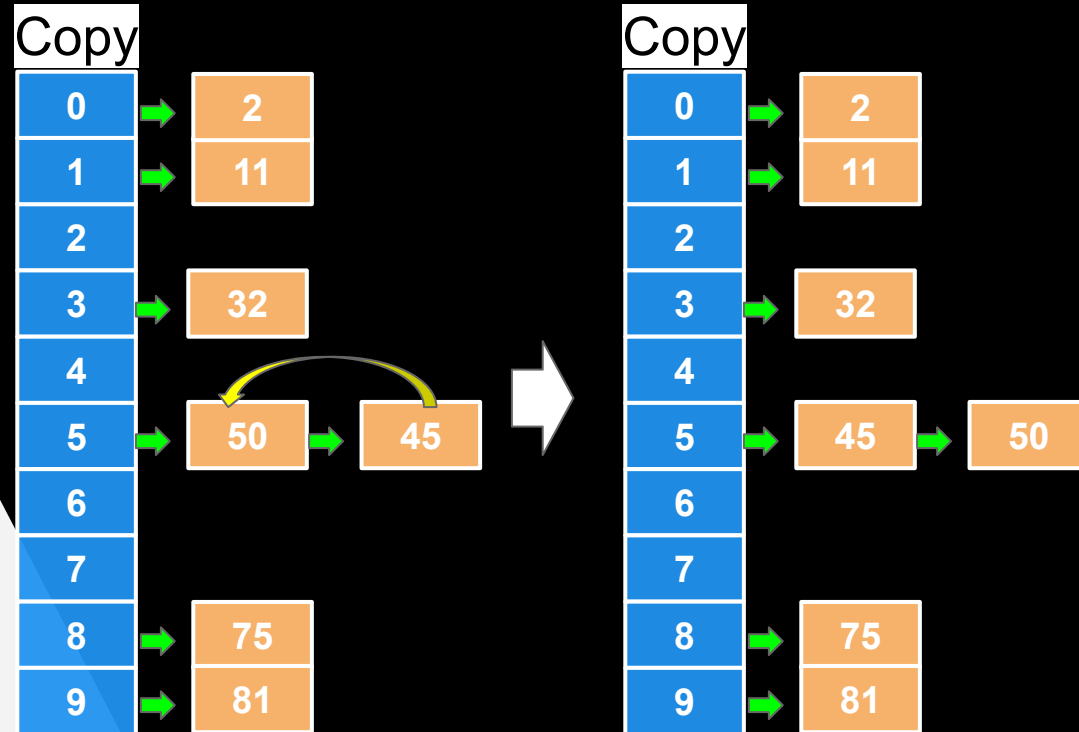
$j = \text{floor}(81/9) \rightarrow j = 9$

$j = \text{floor}(50/9) \rightarrow j = 5$

$j = \text{floor}(32/9) \rightarrow j = 3$

$j = \text{floor}(45/9) \rightarrow j = 5$

Bucket Sort



Bucket Sort

Array []
Elementos

0	1	2	3	4	5	6
2	11	32	45	50	75	81

Final 

Limitations

- ▶ **Mejor:**
 - ▷ Cada cubo es independiente de otros.
- ▶ **Peor:**
 - ▷ Mala distribución de los cubos, generaría trabajo extra sin beneficio .

Complexity and Stability

- ▶ Best case:
 - ▷ $O(n)$
- ▶ Space Complexity:
 - ▷ $O(n)$
- ▶ Worst case:
 - ▷ $O(n^2)$
- ▶ Stability:
 - ▷ Yes

Show me the code



¿ Preguntas ?

1 .¿ Hay una dependencia en la cantidad de buckets creados ?. Explique

2 .¿Cuándo se recomienda realizar un insertion sort?. Explique

¿ Preguntas ?

1 .¿ Hay una dependencia en la cantidad de buckets creados ?. Explique

Si, por la cantidad de elementos en el array, elemento máximo.

2 .¿Cuándo se recomienda realizar un insertion sort?. Explique

Depende del tamaño del contenedor.


```
cout<< " Gracias....!"<<endl;
```

Gracias....!

