



MÉTODOS DE ORDENAMIENTO

# Método Shell



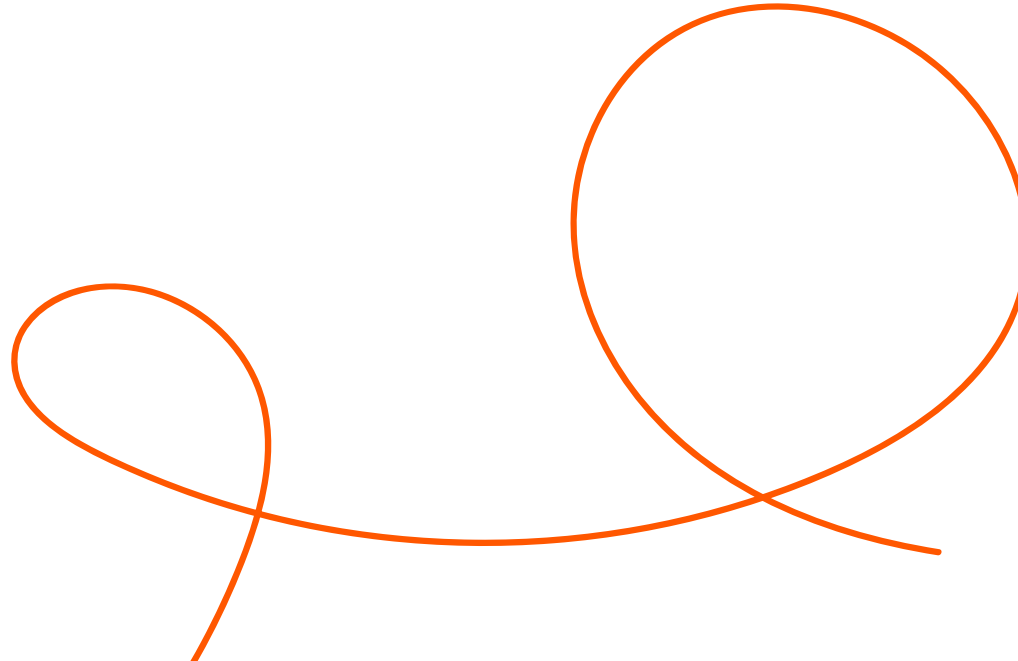
# Shell Sort



Consiste en dividir el arreglo en bloques de varios elementos para organizarlos después por medio del ordenamiento de inserción directa.

El ordenamiento Shell es un algoritmo altamente eficiente basado en la comparación.

En este algoritmo se comienzan a ordenar los elementos más lejanos y gradualmente se va reduciendo la distancia basándose en una secuencia para ordenar todo el arreglo.



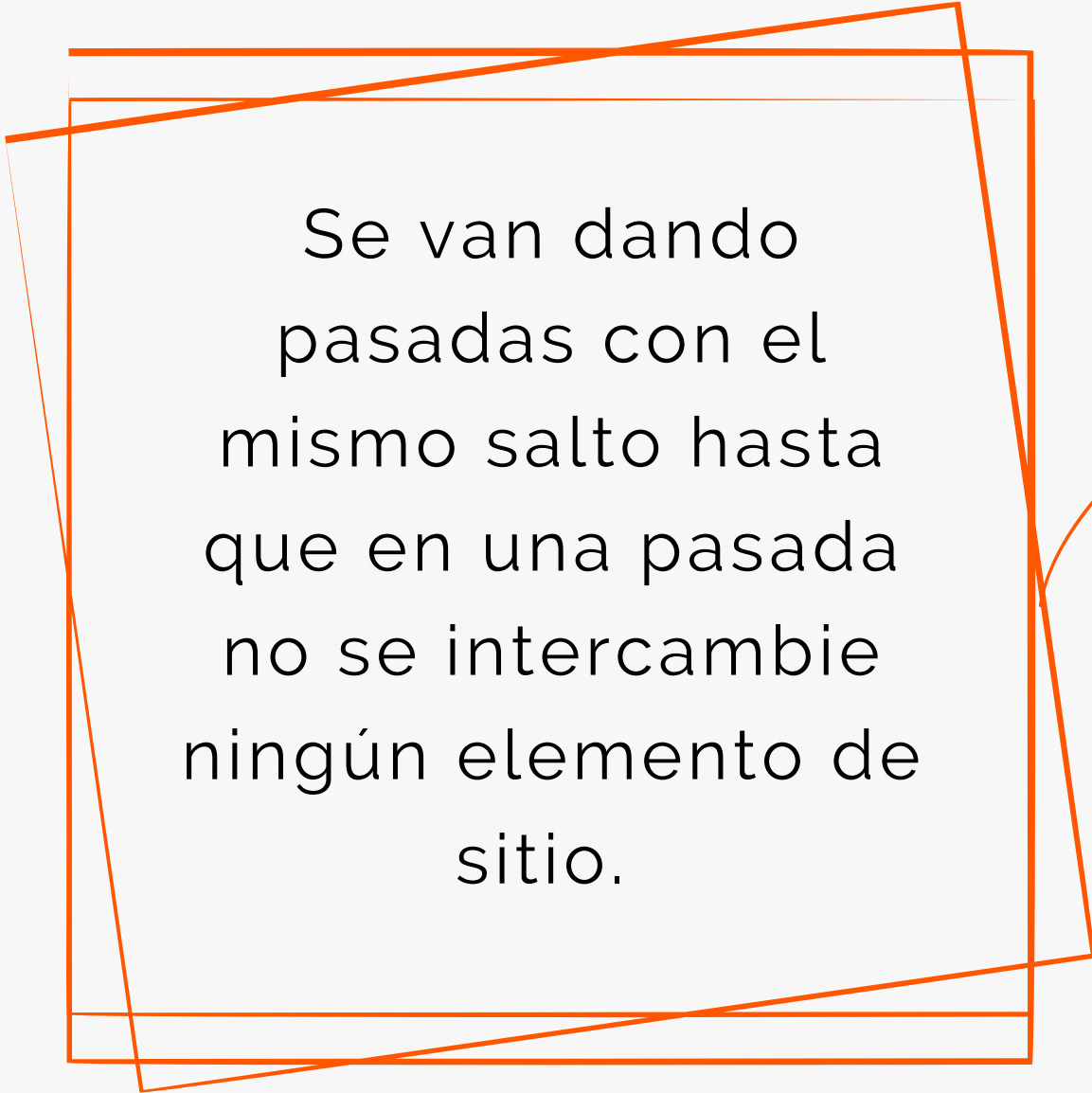


# Proceso

Cualquier algoritmo de ordenación que intercambia elementos adyacentes (como los algoritmos burbuja, selección o inserción) tiene un tiempo promedio de ejecución de orden cuadrático ( $n^2$ ).

El método Shell mejora este tiempo comparando cada elemento con el que está a un cierto número de posiciones llamado salto, en lugar de compararlo con el que está justo a su lado.

Este salto es constante, y su valor inicial es  $N/2$  (siendo  $N$  el número de elementos, y siendo división entera).



Se van dando pasadas con el mismo salto hasta que en una pasada no se intercambie ningún elemento de sitio.

The diagram shows a light gray rectangular area with an orange border. Inside, there are three overlapping orange rectangles. The leftmost rectangle is tilted clockwise. The middle rectangle is horizontal. The rightmost rectangle is tilted counter-clockwise. An orange arrow curves from the right side of the tilted rectangles towards the right text block.

Entonces el salto se reduce a la mitad, y se vuelven a dar pasadas hasta que no se intercambie ningún elemento, y así sucesivamente hasta que el salto vale 1.

# Ventajas

Es un algoritmo muy simple teniendo un tiempo de ejecución aceptable.

Es uno de los algoritmos más rápidos.

No requiere memoria adicional.

Fácil implementación.



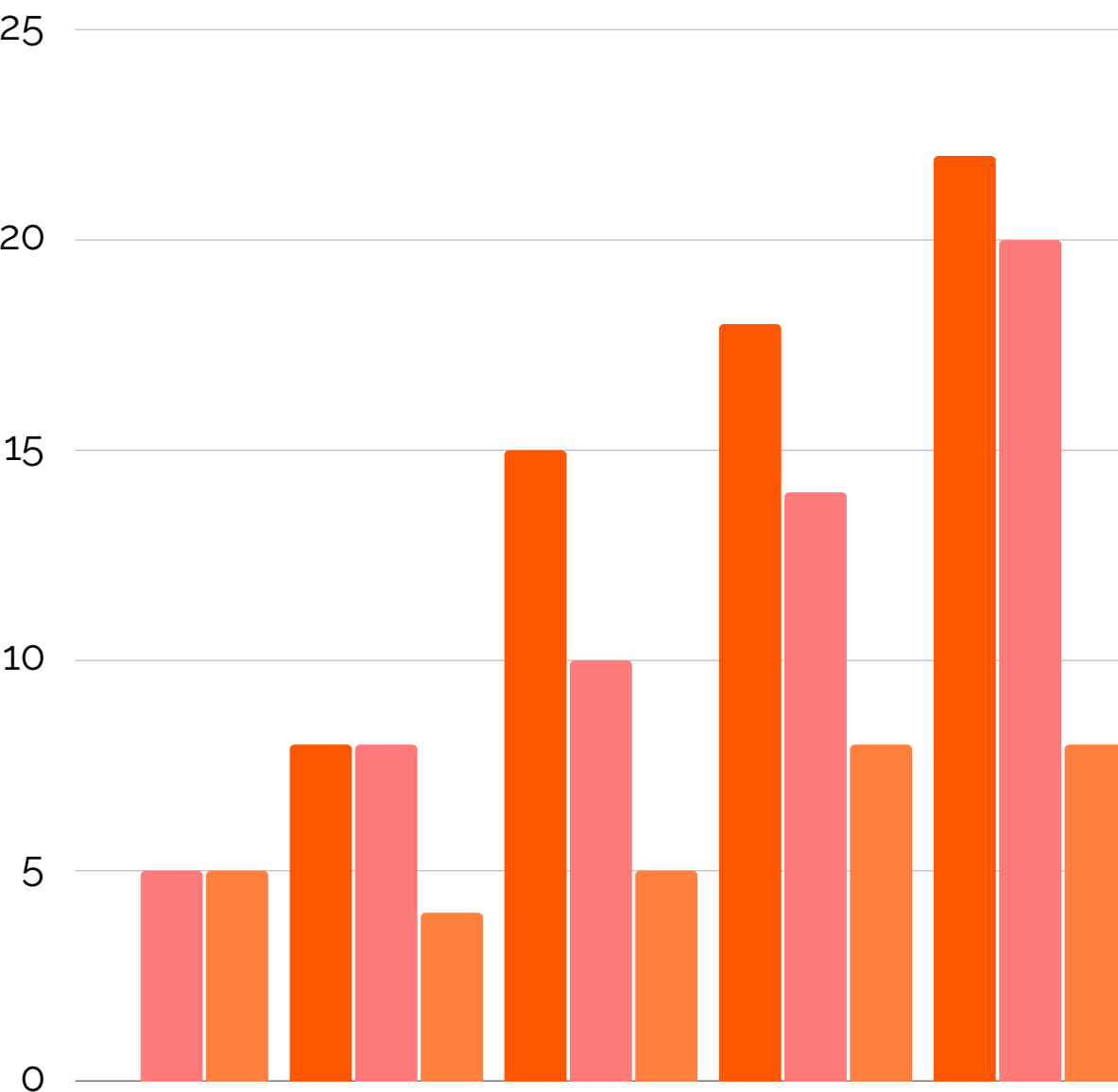
# Desventajas

Su complejidad es difícil de calcular y depende mucho de la secuencia de incrementos que utilice.

Shell Sort es un algoritmo no estable porque se puede perder el orden relativo inicial con facilidad.

Es menos eficiente que los métodos Merge, Heap y Quick Sort.

Realiza numerosas comparaciones e intercambios.



# Algoritmo para clasificación de shell

Paso 1: establezca el valor de  $i$ .

Paso 2: separe la lista en sublistas con el mismo intervalo  $i$ .

Paso 3: utilizando la ordenación por inserción, ordene estas sublistas.

Paso 4: continúe hasta que se haya ordenado toda la lista.

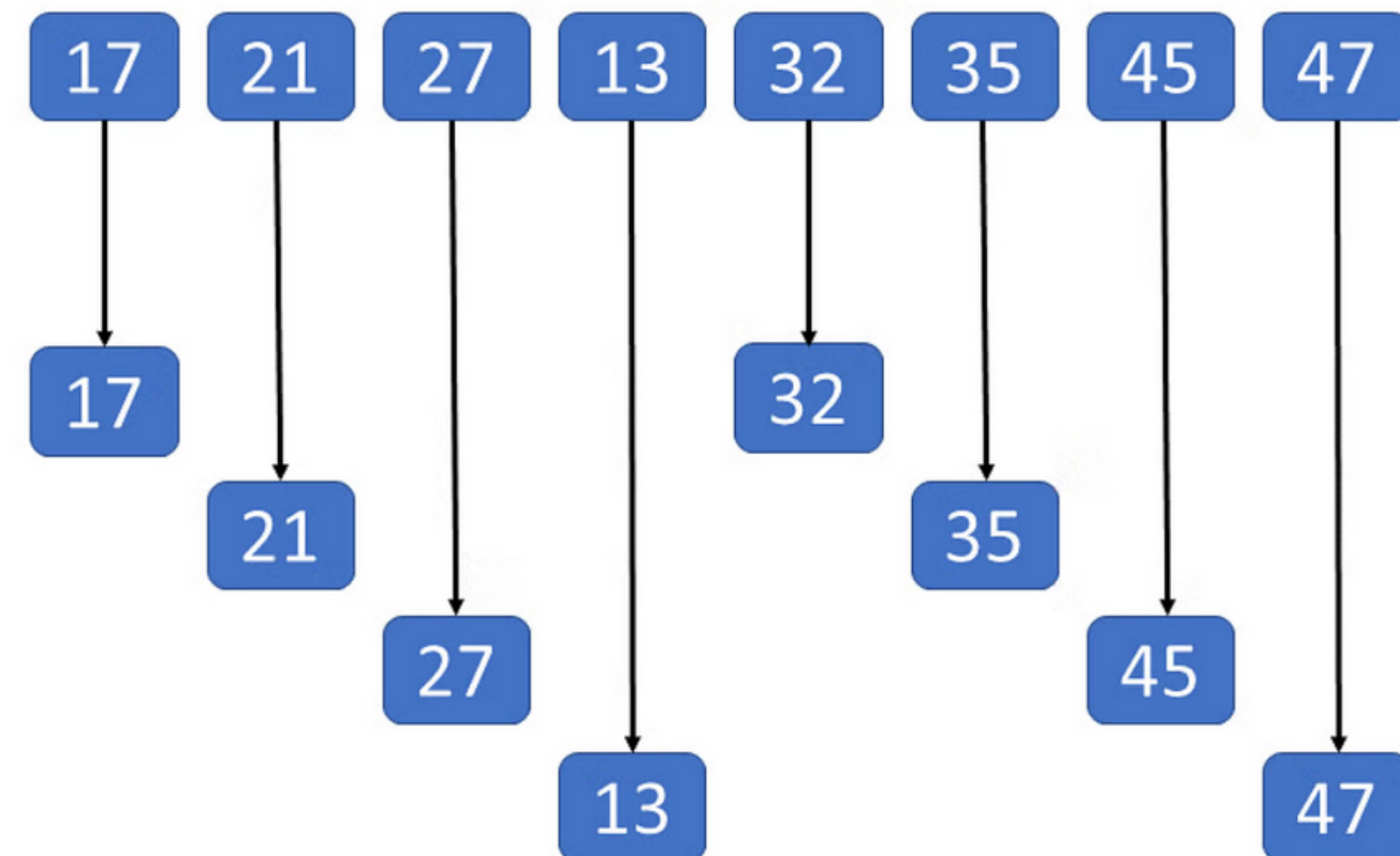


NECESIDADES  
QUE CUBRE

## ¿Cómo funciona un algoritmo de clasificación de shell?

Haga una sublista virtual de todos los valores en el intervalo de cuatro posiciones.

Estos son los valores: (32, 17); (35, 21); (45, 27); y (13, 47).

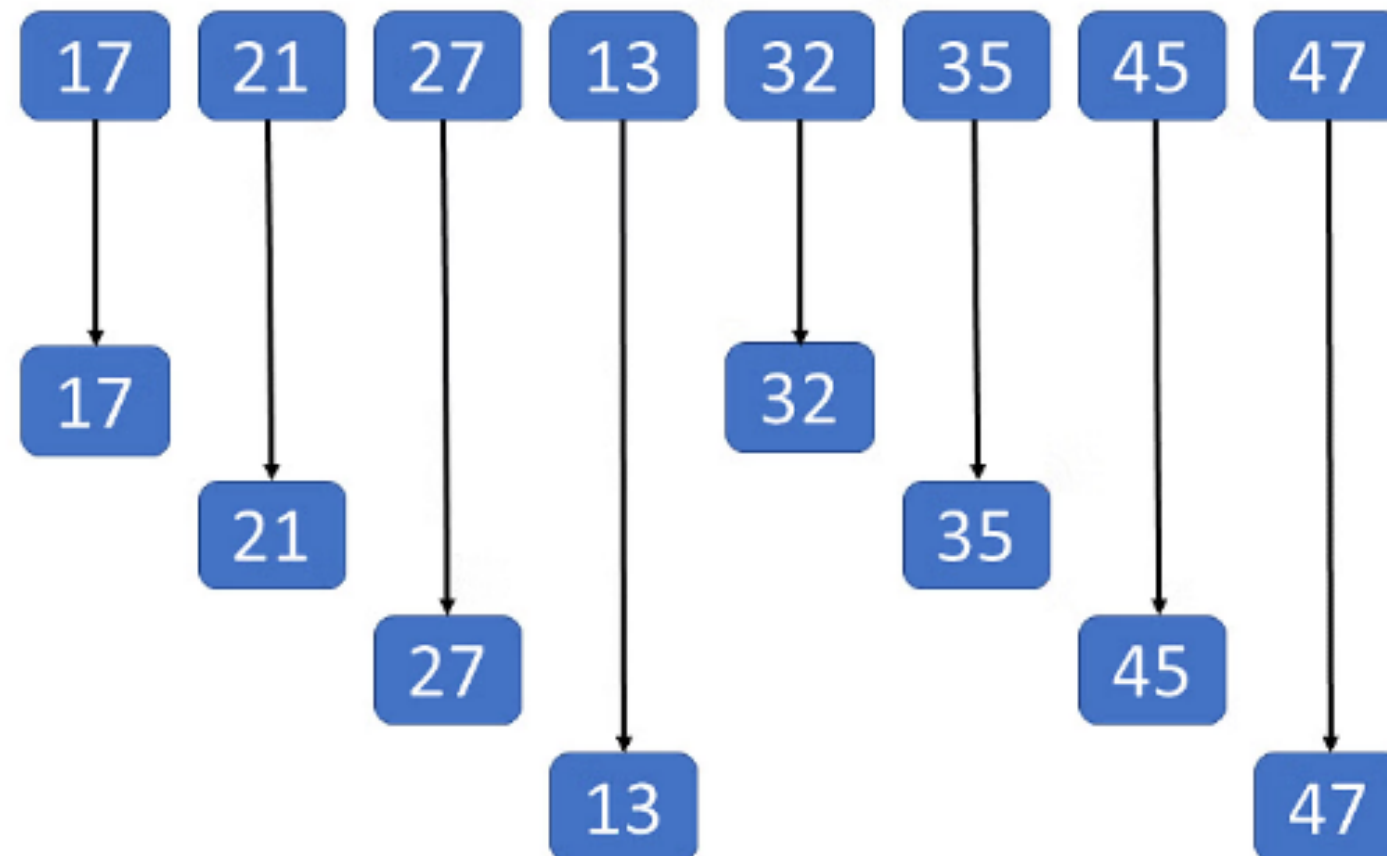




Compara los valores en cada sublista y los intercambia en la matriz original si es necesario. Después de este paso, la nueva matriz debería verse así:



Luego elegimos un intervalo y lo dividimos en dos sublistas: (17, 27, 32, 45) y (21, 13, 35, 47).



Finalmente, necesitas usar el intervalo 1 para ordenar el resto de la matriz.  
La matriz se ordena por ordenación de shell mediante ordenación por inserción.

17	21	27	13	32	35	45	47
17	21	27	13	32	35	45	47
17	21	13	27	32	35	45	47
17	13	21	27	32	35	45	47
13	17	21	27	32	35	45	47

Xavier Alexandro Luviano Grimaldo  
Maria Fernanda Franco Salinas

# Gracias !

22100207  
22100183