

Quicksort Algorithm

Integrantes:

Casas Vazquez Francisco Javier

Flores Guzman Pedro Eduardo

Ordenamiento Rápido (Quicksort)

El ordenamiento rápido (quicksort en inglés) es un algoritmo basado en la técnica de divide y vencerás, que permite, en promedio, ordenar n elementos en un tiempo proporcional a $n \log n$. Esta es la técnica de ordenamiento más rápida conocida. El algoritmo original es recursivo, pero se utilizan versiones iterativas para mejorar su rendimiento (los algoritmos recursivos son en general más lentos que los iterativos, y consumen más recursos).

Charles Antony Richard creador del Algoritmo Quicksort

Quicksort se creó en 1960 por Tony Hoare, un estudiante de visita en la Universidad Estatal de Moscú, en un estudio sobre la traducción automática para el Laboratorio Nacional de Física. Intentando facilitar la búsqueda de palabras en el diccionario Ruso-Inglés existentes almacenados en una cinta magnética.



Charles Antony Richard
Hoare

11 de enero de 1934 (82
años)

Colombo , británica de Ceilán

Descripción del Algoritmo

- Elegir un elemento de la lista de elementos a ordenar, al que llamaremos pivote.
- Resituar los demás elementos de la lista a cada lado del pivote, de manera que a un lado queden todos los menores que él, y al otro los mayores. En este momento, el pivote ocupa exactamente el lugar que le corresponderá en la lista ordenada.
- La lista queda separada en dos sublistas, una formada por los elementos a la izquierda del pivote, y otra por los elementos a su derecha.
- Repetir este proceso de forma recursiva para cada sublista mientras éstas contengan más de un elemento. Una vez terminado este proceso todos los elementos estarán ordenados. Como se puede suponer, la eficiencia del algoritmo depende de la posición en la que termine el pivote elegido.

Eleccion de Pivote

Pivote con el Primer Elemento:

5	3	4	2	6	7	1
---	---	---	---	---	---	---

Pivote con el elemento del centro:

5	3	4	2	6	7	1
---	---	---	---	---	---	---

Pivote con el ultimo Elemento:

5	3	4	2	6	7	1
---	---	---	---	---	---	---

Pivote aleatorio:

5	3	4	2	6	7	1
---	---	---	---	---	---	---

Tiempo de Ejecución

Casos	Tiempo	Descripción
Peor de los casos	$O(n^2)$	Se produce cuando el pivote siempre se inclina hacia un lado, es decir, genera una array de solo un 1 elemento y una segunda con el resto de elementos.
Caso promedio	$O(n \log (n))$	Se produce cuando el pivote se inclina más hacia un lado y los 2 subarrays tienen distinto tamaño de elementos.
Mejor de los casos	$O(n \log (n))$	Se produce cuando el pivote termina en el centro de la lista, dividiéndola en dos sublistas de igual tamaño.

Pseudocodigo

```
para i = 1 hasta 100
    leer(A[i])
Fin para
primero = 1
ultimo = 100
qsort(A[],100)
Fin

Funcion qsort(primero, ultimo:entero)
    i = primero
    j = ultimo
    central = A[(primero,ultimo) div 2]
    repetir
        mientras A[i]central
            j = j - 1
        fin mientras
        si i <= j
            aux = A[i]
            A[j] = A[i]
            A[i] = aux
            i = i + 1
            j = j - 1
        fin si
    hasta que i > j
    si primero < j
        partir(primero,j)
    fin si
    si i < ultimo
        partir(i, ultimo)
    fin si
fin funcion qsort
```

Codigo en C

```
quicksort(a,0,n-1);

void quicksort(double a[ ], int primero, int ultimo){
    int i,j,central;
    double pivote;

    central = (primero + ultimo) / 2;
    pivote = a[central];

    i = primero;
    j = ultimo;

    do{
        while( a[i] < pivote) i++;
        while( a[j] > pivote) j--;

        if( i <= j){
            double tmp;
            tmp = a[i];
            a[i] = a[j];
            a[j] = tmp;
            i++;
            j++;
        }
    }while( i <=j );

    if(primero < j )
        quicksort(a, primero, j);

    if( i < ultimo )
        quicksort(a, i ultimo);
}
```

Ejemplo

8	1	4	9	6	3	5	2	7	0
---	---	---	---	---	---	---	---	---	---

El índice i se detiene en elemento 8 (mayor que el pivote) y el índice j se detiene en el elemento 0 (menor que el pivote)

8	1	4	9	6	3	5	2	7	0
---	---	---	---	---	---	---	---	---	---

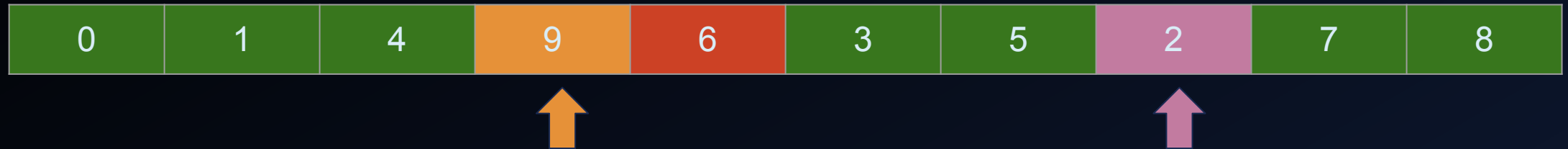


Ahora se intercambian 8 y 0 para que estos dos elementos se sitúen correctamente en cada sublista y se incrementa el índice i y se decrementa j para seguir los intercambios.

0	1	4	9	6	3	5	2	7	8
---	---	---	---	---	---	---	---	---	---



A medida que el algoritmo continúa, i se detiene en el elemento mayor a 9 y j se detiene el elemento menor a 2



Se intercambian los elementos mientras que i y j no se crucen. En el momento que se crucen los índices se detiene el bucle. En el caso anterior se intercambió 9 y 2.



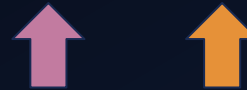
Continúa la exploración y ahora el contador i se detiene en el elemento 6 (que es el pivote) y el índice j se detiene en el elemento menor 5. Se intercambia 6 y 5, se incrementa i y decrementa j .

0	1	4	2	5	3	6	9	7	8
---	---	---	---	---	---	---	---	---	---



Los índices tienen actualmente los valores $i = 5$, $j = 5$. Continúa la exploración hasta que $i > j$, acaba con $i = 6$ y $j = 5$

0	1	4	2	5	3	6	9	7	8
---	---	---	---	---	---	---	---	---	---



Los índices han cruzado sus posiciones en el array. Se detiene la búsqueda por lo que se obtienen sublistas.

0	1	4	2	5	3	6	9	7	8
---	---	---	---	---	---	---	---	---	---

Sublista izquierda

Pivote

Sublista Derecha

Sublista izquierda

0	1	4	2	5	3
---	---	---	---	---	---



0	1	4	2	5	3
---	---	---	---	---	---



0	1	4	2	3	5
---	---	---	---	---	---

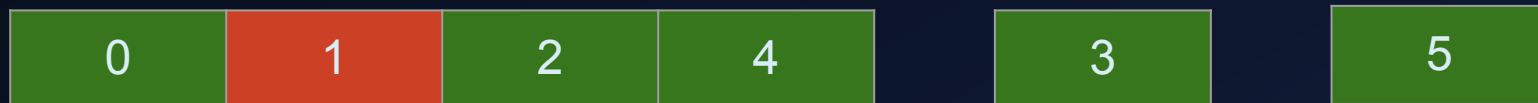
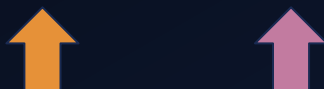
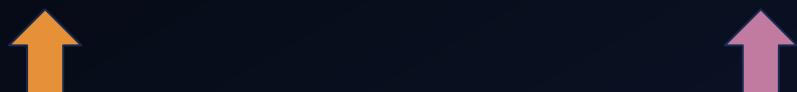


0	1	4	2	3	5
---	---	---	---	---	---

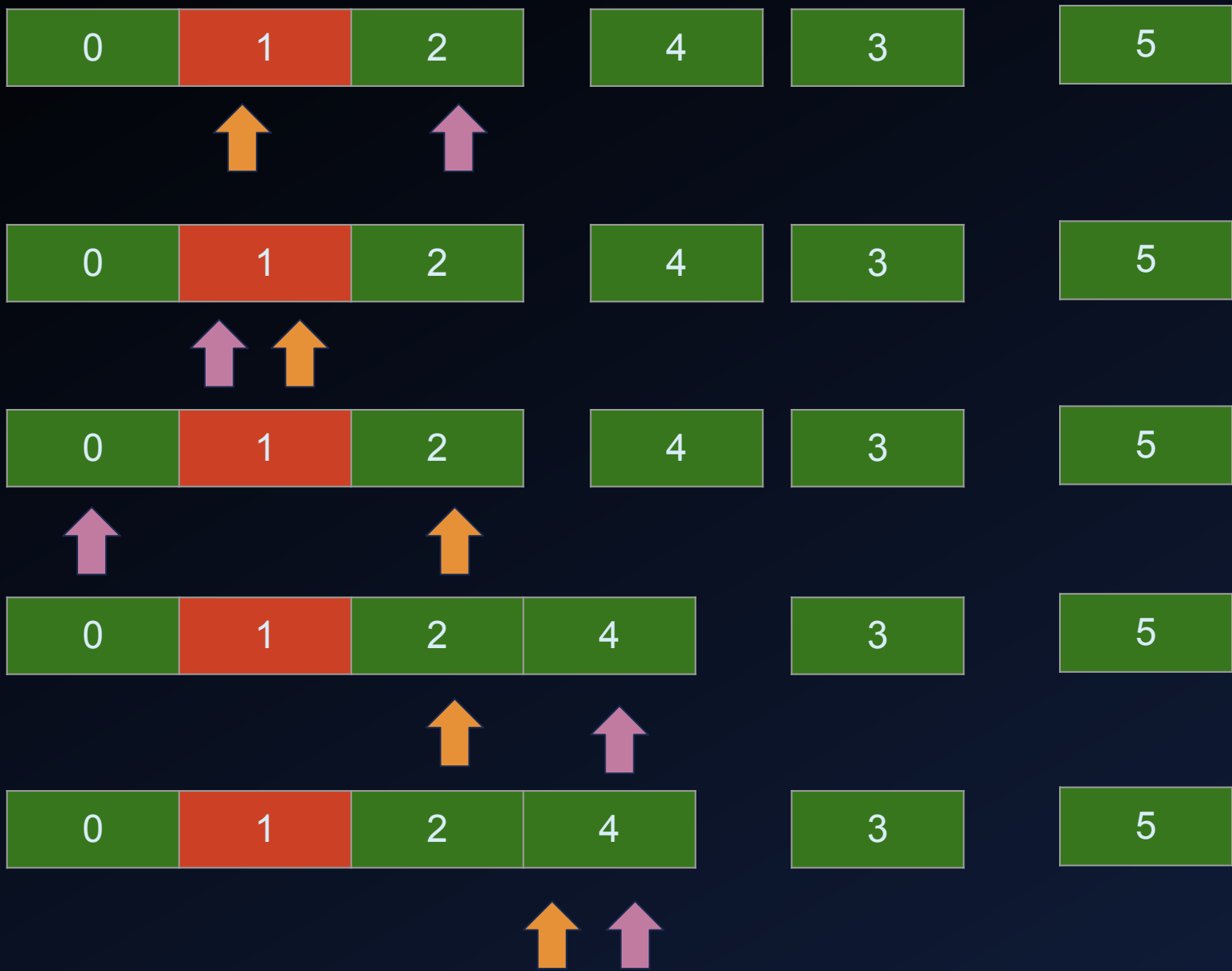


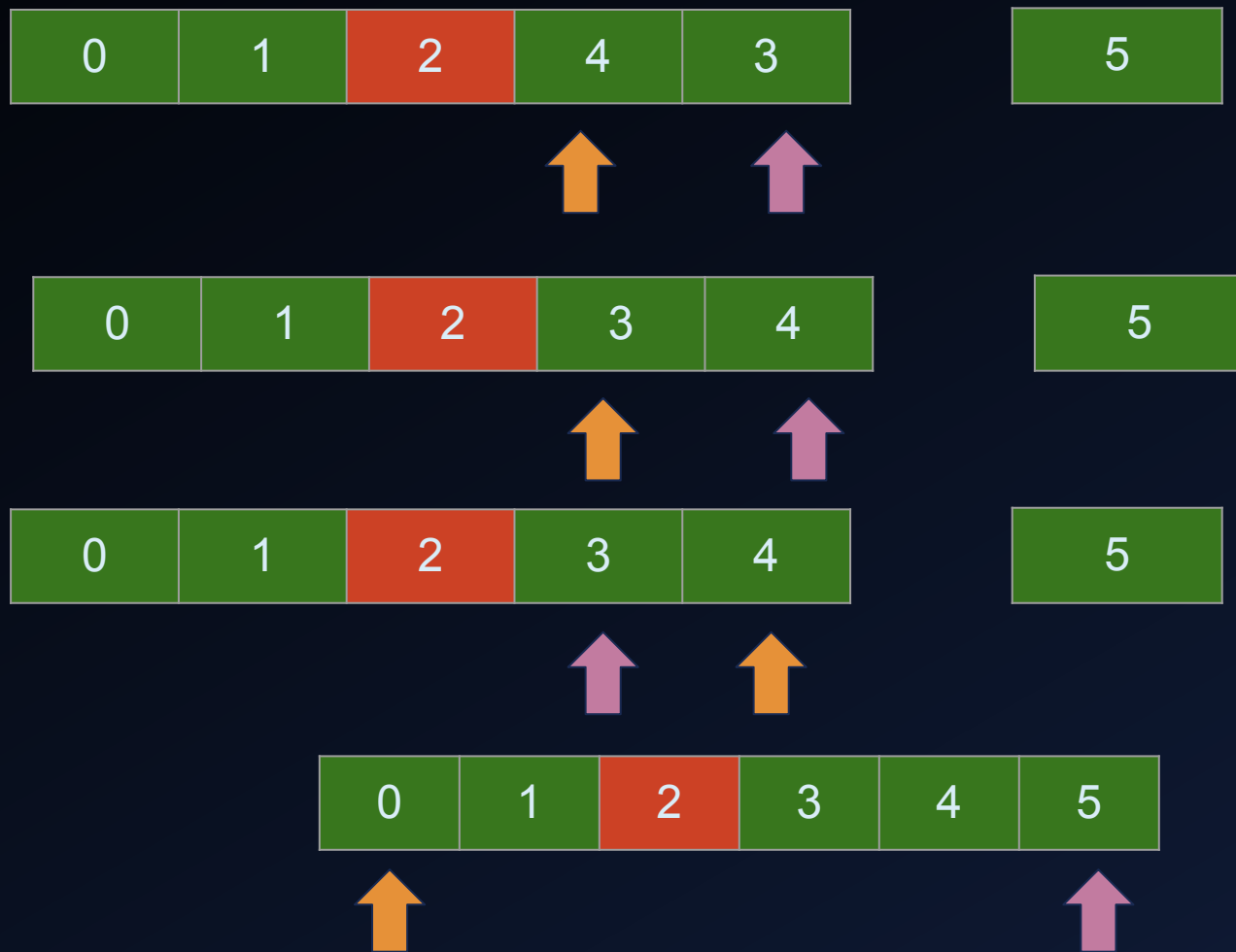
0	1	4	2	3		5
---	---	---	---	---	--	---





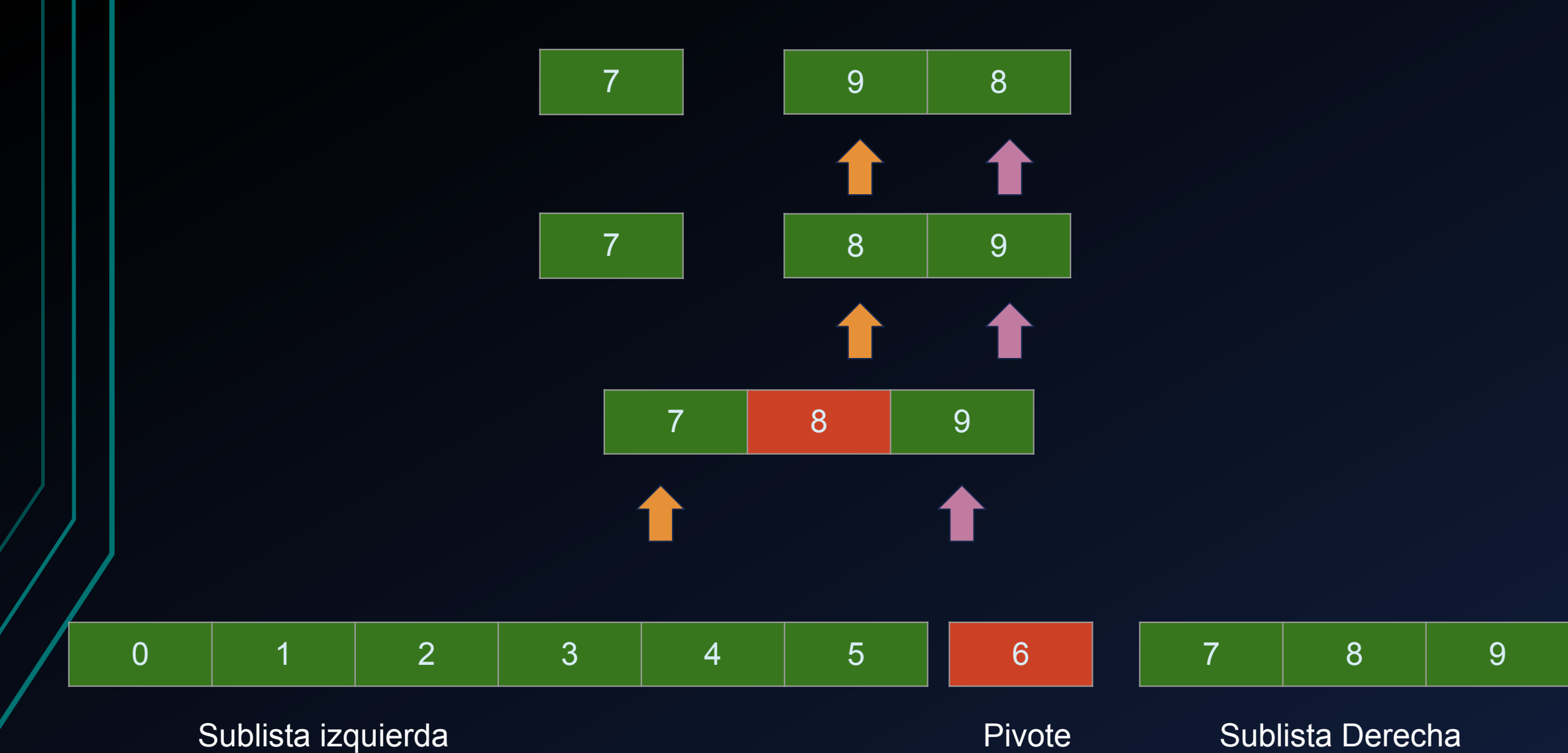






Sublista izquierda





Bibliografia

Libro: Programacion en C Metodología, algoritmos y estructuras de Datos.

Autor:Luis Joyanes Aguilar e Ignacio Zahonero Martinez

Edicion:5ta.

<http://ayudamosconocer.com/significados/letra-q/quicksort.php>