Estructura de Datos

Método de Quicksort

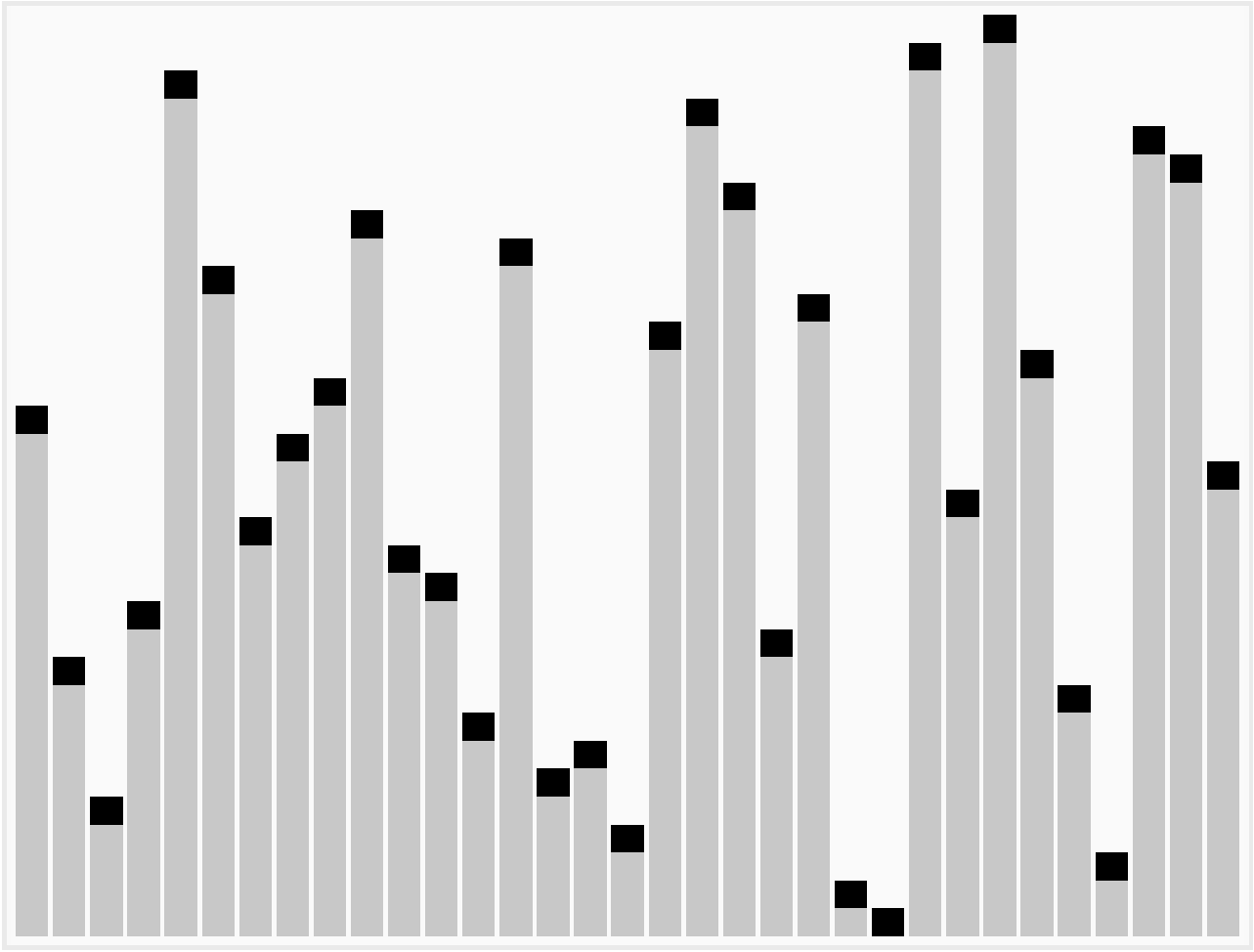
Ordenamiento de Quicksort

El **ordenamiento rápido** 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*.**

Como se puede suponer, la eficiencia del algoritmo depende de la posición en la que termine el pivote elegido.
En el mejor caso, el pivote termina en el centro de la lista, dividiéndola en dos sublistas de igual tamaño. En este caso, el orden de complejidad del algoritmo es **O(n log n)**.

En el peor caso, el pivote termina en un extremo de la lista. El orden de complejidad del algoritmo es entonces de **$O(n^2)$**. El peor caso dependerá de la implementación del algoritmo, aunque habitualmente ocurre en listas que se encuentran ordenadas, o casi ordenadas.

En el caso promedio, el orden es **O(n log n)**.
No es extraño, pues, que la mayoría de optimizaciones que se aplican al algoritmo se centren en la elección del **pivote.**



Algoritmo Método de Quicksort

```
void quicksort(int* izq, int* der) /*Se llama con: quicksort(&vector[0],&vector[n-1]);*/
{
	if(der<izq) return;
	int pivot=*izq;
	int* ult=der;
	int* pri=izq;

	while(izq<der)
	{
		while(*izq<=pivot && izq<der+1) izq++;
		while(*der>pivot && der>izq-1) der--;
		if(izq<der) swap(izq,der);
	}
	swap(pri,der);
	quicksort(pri,der-1);
	quicksort(der+1,ult);
}

void swap(int* a, int* b)
{
	int temp=*a;
	*a=*b;
	*b=temp;
}
```