

Divide y vencerás

Objetivos

- Implementar el algoritmo de ordenación rápida (quicksort) tomando como pivote la mediana de los elementos que ocupan las posiciones primera, última y centro del vector.
- Trabajar en grupo.

Actividad (Equipos de 2 o 3 alumnos)

El algoritmo de ordenación rápida (quicksort) es un algoritmo que sigue una estrategia divide y vencerás para ordenar vectores. Una implementación de este algoritmo es:

```
función quicksort(V:&tipo[n], izq:natural U {0}, der:natural U {0})
    i,d,p:natural U {0}
    pivote:tipo

    p ← seleccionar_pivote(V,izq,der) // 0 si no pivote
    si p > 0
        pivote ← Vp
        i ← izq
        d ← der
        mientras i ≤ d hacer
            mientras Vi < pivote hacer
                i ← i + 1
            fmientras
            mientras Vd > pivote hacer
                d ← d - 1
            fmientras
            si i ≤ d
                intercambiar(V,i,d)
                i ← i + 1
                d ← d - 1
            fsi
        fmientras
        si izq < d quicksort(V,izq,d) fsi
        si i < der quicksort(V,i,der) fsi
    fsi
ffunción
```

1.- Implementad este algoritmo quicksort en C++ siguiendo las reglas de transformación de pseudocódigo vistas en la práctica 1. Cread un programa que pida por teclado el tamaño del vector y los elementos del mismo. Una vez introducidos los valores se imprimirá el vector antes de ordenarlo y después de ordenarlo utilizando este algoritmo quicksort. Como estrategia de selección del pivote se utilizará la mediana de los elementos que se encuentran en las posiciones primera, última y centro del vector. Esta selección se realizará dentro de la función seleccionar_pivote.

2.- Implementad una nueva función llamada quicksort_traza que además de ordenar el vector realice la traza del algoritmo imprimiendo la secuencia de llamadas recursivas que se producen. En cada llamada se indicará el número de llamada, el valor de los parámetros izq y der, la posición del pivote y el pivote seleccionado. Podéis añadir a la función los parámetros que consideres necesarios para producir el formato de salida especificado. Modificad el programa creado en la actividad 1 para incluir una llamada a este algoritmo en lugar de al algoritmo quicksort original. En la diapositiva 60 del Tema 3 tenéis un ejemplo de traza para comprobar el resultado de vuestra traza. Ejecutar el programa para más vectores y comprobar que es correcto.