

## RELACIÓN DE PROBLEMAS Divide y Vencerás

### Problema 1: Algoritmos de ordenación: Mergesort

Implementa el algoritmo de ordenación Mergesort sobre vectores de enteros de tamaño  $n$ .

- Implementar en primer lugar el algoritmo de ordenación de la burbuja mejorado, para utilizarlo cuando el tamaño del caso sea menor o igual que el umbral.
- ¿Cuál es la eficiencia del algoritmo mergesort implementado?
- ¿Cuál es el mejor umbral?

**NOTA:** Para medir tiempos de ejecución utiliza la biblioteca [time.h](#) del lenguaje de programación C.

### Problema 2: Búsqueda ternaria

Implementar el algoritmo de la búsqueda ternaria sobre un vector de enteros ordenado de tamaño  $n$ .

Dicho algoritmo es similar a la búsqueda binaria, pero en vez de dividir el vector en dos partes iguales y decidir si se continúa la búsqueda por la parte izquierda o por la derecha, este nuevo algoritmo divide el vector en tres partes iguales y decide si tiene que buscar en la parte izquierda, en la central o en la derecha.

- Implementar en primer lugar el algoritmo de búsqueda secuencial, para utilizarlo cuando el tamaño del caso sea menor o igual que el umbral.
- ¿Cuál es la eficiencia del algoritmo de búsqueda ternaria implementado?
- ¿Cuál es el mejor umbral?

### Problema 3: Elemento Mayoritario de un Conjunto de Datos

Dado un conjunto  $C$  de  $n$  elementos (no necesariamente ordenables) se dice que un elemento  $x$  es mayoritario en  $C$  cuando el número de veces que aparece  $x$  en  $C$  es estrictamente mayor que  $n/2$ . Dado un conjunto de elementos se desea saber si un conjunto contiene un elemento mayoritario y devuelva tal elemento cuando exista.

- ¿Cómo sería el algoritmo clásico que se podría diseñar para resolver este problema? ¿Cuál sería su eficiencia?
- Diseñar e implementar un algoritmo basado en la técnica Divide y Vencerás para solucionar este problema. ¿Cuál sería su eficiencia?

## Problema 4: Suma de los elementos de un vector, excepto el mayor y el menor

Sea T un vector desordenado de números enteros de tamaño n. Se desea calcular la suma de todos sus elementos a excepción del mayor y del menor. Por ejemplo, para el siguiente vector de 8 elementos la suma es 30.

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

- Diseñar e implementar el algoritmo clásico.
- Diseñar e implementar el Divide y Vencerás.
- ¿Cuál es el mejor umbral?

## Problema 5: Calcular el número de elementos que tiene una matriz entre dos valores dados.

En una matriz nxn, con n potencia de 2, se almacenan una serie de números enteros positivos de forma desordenada. Se desean conocer, cuántos números hay en la matriz cuyo valor está en el intervalo cerrado [a, b], siendo a y b dos valores pasados como parámetros. Por ejemplo, para la siguiente matriz, si a=5 y b=8, el resultado debe ser 7.

4	8	5	6
6	12	4	1
3	1	3	5
5	13	6	2

- Diseñar e implementar el algoritmo clásico.
- Diseñar e implementar el Divide y Vencerás.

## Problema 6: Carreras de líquidos

Se ha celebrado una carrera de transporte de líquidos. En dicha carrera todos los participantes parten con un recipiente encima de la cabeza que tiene una capacidad C, la misma para todos. Cuando cada uno de los participantes va llegando a la meta, entrega su recipiente y se anota el dorsal y la cantidad de líquido que han conseguido transportar. Por ejemplo, si hay 10 participantes y se mide en mililitros, los valores anotados podrían ser los siguientes (dorsal y cantidad entregada):

07	03	10	01	04	06	08	02	05	09
575	842	27	135	970	10	1000	624	468	580

Para la entrega de premios se desea conocer el dorsal del corredor que ocupa la posición k (valor entre 1 y el número de participantes n) en función de la cantidad de líquido entregada. Diseñar un algoritmo basado en la técnica Divide y Vencerás para resolver este problema. NOTA: No se puede hacer una ordenación total de los participantes en función de la cantidad de líquido entregada.