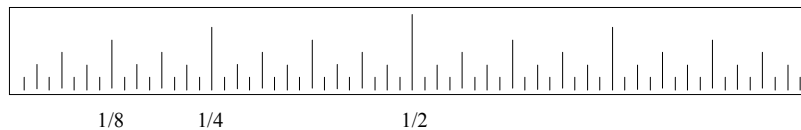


ANÁLISIS Y DISEÑO DE ALGORITMOS I

Práctico N° 5. Divide y Conquista

Implemente algoritmos en C++ utilizando la técnica Divide y Conquista para resolver los siguientes problemas. Determine para cada uno su **complejidad temporal**.

1. Determinar si una secuencia de n caracteres está ordenada alfabéticamente.
2. Dado un arreglo de números naturales ordenados crecientemente, implemente un algoritmo que determine si un número dado pertenece al arreglo.
3. Determinar los valores mínimo y máximo en un arreglo de números reales.
4. Dado un arreglo de números, ordene los elementos implementando el método de ordenamiento *Merge sort*.
5. Dado un arreglo de números, ordene los elementos implementando el método de ordenamiento *Quicksort*.
6. Dadas dos imágenes en blanco y negro sobre una misma región representadas con Quadtree, realizar:
 - a. UNIÓN de las dos imágenes
 - b. INTERSECCIÓN de las dos imágenes
7. Consideremos la tarea de dibujar las marcas para cada pulgada sobre una regla: una marca en la mitad de la pulgada, marcas ligeramente más cortas a intervalos de $1/4''$, marcas un poco más cortas aún a $1/8''$, etc., como se muestra en la siguiente figura:



Si la resolución deseada es $1/2^n$ implemente un algoritmo que dibuje una marca en cada punto entre 0 y 2^n , incluidos los extremos.

8. Dadas dos matrices A y B resolver su producto por el algoritmo de Strassen suponiendo:
 - a. Las matrices A y B son cuadradas completas
 - b. Las matrices A y B son ralas.
9. Sea $a[1..n]$, $n \geq 1$, un vector de enteros diferentes y ordenados crecientemente, tal que algunos de los valores pueden ser negativos. Diseñar un algoritmo que devuelva un índice natural k , $1 \leq k \leq n$, tal que $a[k] = k$, siempre que tal índice exista.