



## **Práctica 2: Divide y Vecenrás**

**Algoritmos básicos**

**Alicatar**

Dpto. Ciencias de la Computación e Inteligencia Artificial  
E.T.S. de Ingenierías Informática y de Telecomunicación  
Universidad de Granada



**DECSAI**



**Algorítmica**

Grado en Ingeniería Informática D

## Índice de contenido

1.Introducción.....	3
2.Algoritmos básicos.....	3
2.1.Máximo y Mínimo.....	3
2.2.Moda de un conjunto de enteros.....	3
3.Como seguir con la Práctica.....	4
4.Tornillos con sus tuercas.....	4
4.1.Programa Tornillos.....	4
5.Enlosado de un Cuarto de Baño.....	4
5.1.Descripción del Problema.....	4
5.2.Descripción de la técnica divide y vencerás.....	5
5.3.Programa Alicatar.....	6
5.4.Eficiencia.....	7
6.Material asociado.....	7
7.Práctica a entregar.....	7



# 1. Introducción

Los objetivos de este guión de prácticas son los siguientes:

1. Entender la técnica de diseño de algoritmos Divide y Vencerás DyV . Hacer uso de esta técnica para resolver un problema
2. Implementar algoritmos básicos usando esta técnica, y otros algoritmos más complejos.
3. Seguir practicando con la obtención de la eficiencia de algoritmos.

## 2. Algoritmos básicos.

### 2.1. Máximo y Mínimo

Dado un conjunto de enteros diseñar un algoritmo DyV para encontrar el máximo y el mínimo. Para ello implementar un programa que genere un vector con un determinado número de enteros y obtener el máximo y mínimo de este conjunto. Para ello debéis implementar la función:

***pair<int ,int> Max\_Min(const vector<int> & v);***

1. Obtener la eficiencia teórica y empírica del algoritmo.
2. Dar un nivel entre [0,10] del nivel de dificultad de como has percibido este ejercicio. Para ello **0** es igual a no tiene ningún nivel de dificultad y **10** tiene mucha dificultad

### 2.2. Moda de un conjunto de enteros

Dado un conjunto de enteros diseñar un algoritmo DyV para encontrar el elemento moda del conjunto. El elemento moda es aquel que aparece con más frecuencia.

1. Obtener la eficiencia teórica y empírica del algoritmo.
2. Dar un nivel entre [0,10] del nivel de dificultad de como has percibido este ejercicio. Para ello **0** es igual a no tiene ningún nivel de dificultad y **10** tiene mucha dificultad

**IDEA.-** Definido un elemento pivote, dividir el conjunto en tres conjuntos:

1. Conjuntos de elementos menores y diferentes al pivote. Este conjunto se almacena en Heterogeneos.
2. Conjuntos de elementos igual al pivote. Este conjunto se almacena en Homogeneos
3. Conjunto de elementos mayores y diferentes al pivote. Este conjunto se almacena en Heterogeneos.

**Con estas definiciones los pasos a seguir son:**

1. Sea **He** el conjunto en heterogeneos de mayor cardinalidad (el conjunto más grande).
2. Sea **Ho** el conjunto de homogeneos de mayor cardinalidad (el conjunto más grande)
3. Si Ho tiene más elementos que He entonces hemos terminado y la moda es el elemento que aparece en Ho.
4. En otro caso volver a dividir según 1-3 He.

### 3. Como seguir con la Práctica.

Si al hacer el ejercicio 1-2 tu percepción de dificultad es mayor a 6 debes hacer el ejercicio 4 en otro caso debes hacer el ejercicio 5.

### 4. Tornillos con sus tuercas.

Se dispone de un conjunto de  $n$  tornillos de diferente tamaño y sus correspondientes  $n$  tuercas, de forma que cada tornillo encaja perfectamente con una y sólo una tuerca. Dado un tornillo y una tuerca, uno es capaz de determinar si el tornillo es menor que la tuerca, mayor, o encaja exactamente. Sin embargo, no hay forma de comparar dos tornillos o dos tuercas entre ellos para decidir un orden. Se desea ordenar los dos conjuntos de forma que los elementos que ocupan la misma posición en los dos conjuntos emparejen entre sí

IDEA: Se procederá mediante un algoritmo Divide y Vencerás de la siguiente forma. Representaremos los conjuntos de tuercas y tornillos mediante dos vectores de tamaño  $n$  que contienen las tuercas (tamaños) y los tornillos (tamaños). El algoritmo comienza escogiendo un tornillo aleatoriamente, este será el pivote para dividir los tuercas en tres conjuntos:  $S_1$  será el conjunto de tuercas que tienen un tamaño menor al tornillo ;  $S_2$ : el conjunto de tuercas que tienen igual tamaño igual al tornillo; y  $S_3$  el conjunto de tuercas que tienen tamaño mayor que la del tornillo. A continuación usando la tuerca que encaja perfectamente con el tornillo elegido, realizamos una partición similar en el conjunto de los tornillos dividiéndolo en tres conjuntos (aquellos tornillos menores que la tuerca, el tornillo elegido inicialmente que encaja con la tuerca y los tornillos mayores que la tuerca). El algoritmo, de forma recursiva, aplica el mismo procedimiento a cada uno de los dos grupos que se han formado, es decir, el de los tornillos y tuercas menores que el par encontrado, y el de los mayores.

1. Obtener la eficiencia teórica y empírica del algoritmo.
2. Dar un nivel entre  $[0,10]$  del nivel de dificultad de como has percibido este ejercicio. Para ello **0** es igual a no tiene ningún nivel de dificultad y **10** tiene mucha dificultad

#### 4.1. Programa Tornillos

Este programa obtiene el emparejamiento de tornillos y tuercas. Un ejemplo de llamada a este programa desde la línea de órdenes:

```
prompt% tornillos 1000
```

Los parámetros de entrada son los siguientes:

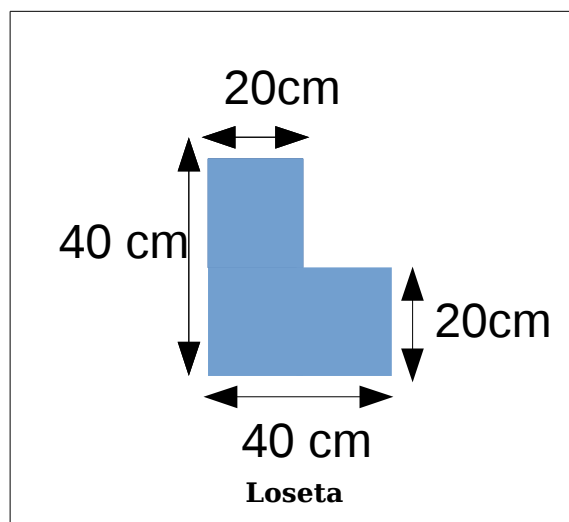
1. El número de tuercas y tornillos que debe generar aleatoriamente.

El programa deberá generar aleatoriamente 1000 tornillos, a continuación con estos tamaños de tornillos debe generar una permutación (ordenar de forma aleatoria) para obtener el conjunto de tuercas, también 1000.

### 5. Enlosado de un Cuarto de Baño

#### 5.1. Descripción del Problema

A un maestro albañil de reconocido prestigio le encomiendan la tarea de enlosar los cuartos de baño de un edificio en construcción. Para ello, y por motivos estéticos, le piden que utilice un determinado tipo de losa en forma de 'L'.



El albañil debe tener en cuenta que, al enlosar cada cuarto de baño deberá dejar el hueco del desagüe sin cubrir. El tamaño de cada cuarto de baño y la ubicación del desagüe puede variar, aunque siempre se cumple la condición de que el suelo es cuadrado con un tamaño de lado de  $2^n \cdot 20$  cm, y el desagüe tiene un tamaño de  $20 \text{ cm} \times 20 \text{ cm}$ .

Bajo estas condiciones, el albañil sabe que siempre podrá cubrir el suelo (dejando el hueco del desagüe) utilizando losas enteras sin necesidad de partir ninguna y, por supuesto, sin superponerlas. Por tanto, se propone diseñar una forma sistemática para ubicar las losas según el procedimiento descrito en la siguiente sección.

En la Figura 1 se muestra un ejemplo de enlosado. El cuadrado negro en el centro representa el desagüe.

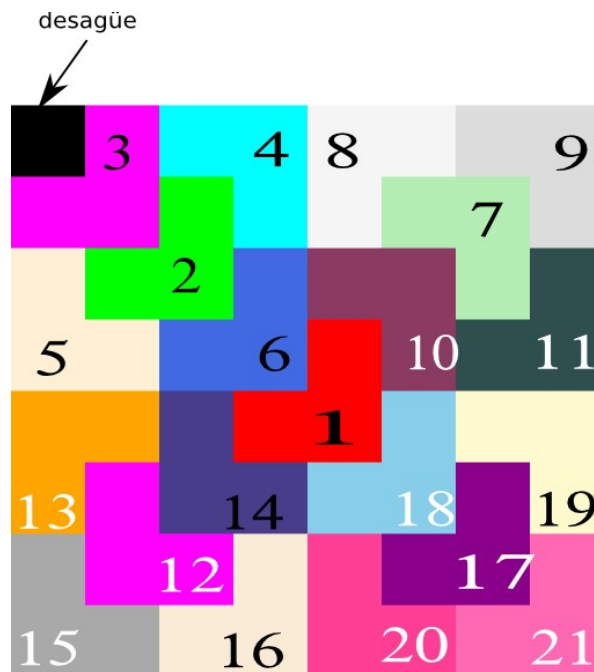
## 5.2. Descripción de la técnica divide y vencerás

Dado un problema de tamaño  $2^n \times 2^n$  casillas, donde hay una casilla que debe quedar libre, se divide en cuatro subproblemas de tamaño  $2^{n-1} \times 2^{n-1}$  dejando una casilla libre en cada uno de ellos. La división puede hacerse simplemente partiendo por la mitad (horizontal y verticalmente) la cuadrícula. En esta división, esta claro que uno de los cuatro subproblemas contendrá la casilla que debe quedar libre. Para hacer que los otros tres subproblemas tengan también una casilla que no debe cubrirse, bastará con ubicar una losa en forma de 'L' que cubra los tres subproblemas. Esta losa se colocará en el centro y formará parte de los tres subcuadrantes donde no está el desagüe. En la figura 6 se ilustra un ejemplo de división. Con ello, hemos conseguido reducir el problema original a cuatro de menor tamaño, que podemos abordar recursivamente de la misma forma. Este proceso se continua hasta llegar al caso trivial del problema de tamaño  $2 \times 2$  con

*Figura 1: Ejemplo de Enlosado. En el centro, en negro, el desagüe. En este ejemplo el cuarto de baño tiene dimensiones  $2^3 \times 2^3$*



una casilla ocupada, donde basta con ubicar la losa en las otras tres casillas sin cubrir.



*Figura 2 Ejemplo de enlosado de un cuarto de baño con dimensión  $2^2 \times 2^2$ . Los números sobre las losetas indican el orden en que fueron puestas.*

### 5.3. Programa Alicatar

Este programa obtiene el alicatado del suelo de un cuarto de baño con un desagüe. Un ejemplo de llamada a este programa desde la línea de órdenes:

```
prompt% alicatar 8 0 0 paleta.txt res_8_8_0_0.ppm
```

Los parámetros de entrada son los siguientes:

1. Dimensión del cuarto de baño (filas y columnas adoptarán este valor). Debe ser potencia de 2.
2. Fila donde se sitúa el desagüe
3. Columna donde estará el desagüe
4. Fichero con la paleta de colores para obtener una representación gráfica del alicatado
5. Imagen resultante

Además en el terminal aparecerá los valores de la matriz que codifican el enlosado como se muestra a continuación:

0	3	4	4	8	8	9	9
3	3	2	4	8	7	7	9
5	2	2	6	10	10	7	11
5	5	6	6	1	10	11	11
13	13	14	1	1	18	19	19
13	12	14	14	18	18	17	19
15	12	12	16	20	17	17	21
15	15	16	16	20	20	21	21

## 5.4. Eficiencia

El alumno aportará un análisis de la eficiencia del programa. Prestando mayor atención a la función que implementa la técnica Divide y Vencerás. Para ello se hará un análisis de la eficiencia tanto teórica como empírica. Se aportará un documento pdf donde se detalle el cálculo de la eficiencia teórica y gráficas que muestren la eficiencia empírica.

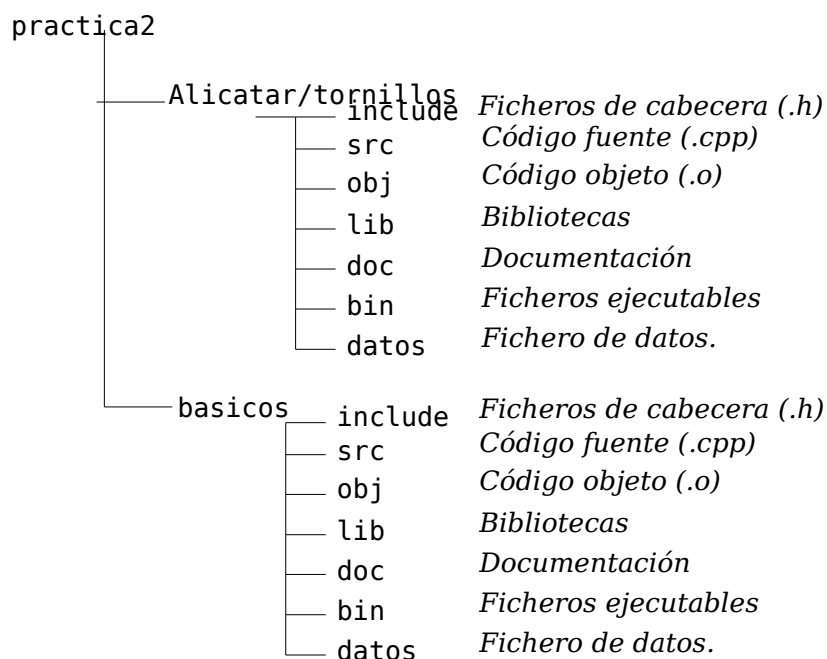
## 6. Material asociado

Para llevar a cabo la práctica el alumno tiene a su disposición tres módulos: imagen (imagen.h, imagen.cpp ), imageES (imagenES.h y imagenES.cpp), y paleta, con objeto de obtener los resultados usando las imágenes para el ejercicio 5.

## 7. Práctica a entregar

El alumno deberá empaquetar todos los archivos relacionados en el proyecto en un archivo con nombre “practica2.tgz” y entregarlo antes de la fecha que se publicará en la página web de la asignatura. Tenga en cuenta que no se incluirán ficheros objeto, ni ejecutables, ni la carpeta datos. Es recomendable que haga una “limpieza” para eliminar los archivos temporales o que se puedan generar a partir de los fuentes. Para aquellos alumnos que decidan hacer el ejercicio 5 desarrollarlo en la carpeta alicatar. Para aquellos otros alumnos que quieran hacer el ejercicio 4 desarrollarlo en la carpeta tornillos.

El alumno debe incluir el archivo *Makefile* para realizar la compilación. Tenga en cuenta que los archivos deben estar distribuidos en directorios:



Para realizar la entrega, en primer lugar, realice la limpieza de archivos que no se incluirán en ella, y sitúese en la carpeta superior (en el mismo nivel de la carpeta “*practica2*”) para ejecutar:

```
prompt% tar zcv practica2.tgz practica2
```

*tras lo cual, dispondrá de un nuevo archivo practica2.tgz que contiene la carpeta practica2 así como todas las carpetas y archivos que cuelgan de ella.*