



Práctica 2: Búsqueda con retroceso

1. CONSIDERACIONES GENERALES

- La entrega de la práctica se realizará en *hendrix* utilizando el siguiente comando:

```
hendrix:> someter ab_23 practica2.tar
```

- Se entregará un fichero **practica2.tar** que contiene un directorio denominado **practica2_NIA1_NIA2** (siendo NIA1 y NIA2 los números identificadores de cada estudiante asignados por la Universidad de Zaragoza, y NIA1 será el NIA menor. En el caso de un grupo de práctica formado por un único alumno, el directorio tendrá como nombre **practica2_NIA** (con el identificador de ese alumno).
- El directorio incluirá los siguientes ficheros de texto:
 - Descripción general del directorio: cómo está organizado, instrucciones de instalación, compilación y ejecución, instrucciones para repetir las pruebas, etc. (tiene que llamarse **LEEME**).
 - Listados del código debidamente comentados. Deberán seguir una estructura lógica para poder encontrar y navegar adecuadamente cada una de las partes de la práctica.
 - Un programa para la *shell* denominado **ejecutar.sh** que automatice la compilación y ejecución de los programas entregados con los casos de prueba.
 - Los ficheros auxiliares de entrada necesarios para ejecutar las pruebas del punto anterior.
- El directorio incluirá también un informe con la presentación y análisis de resultados (fichero PDF, máximo 3 páginas sin portada). Indicar: nombre, apellidos y NIA de cada miembro del grupo de práctica.
- Fechas límite de entrega para la primera convocatoria:**

Grupo	Fecha y hora
Jueves A	17/04/2024 8:00AM
Viernes A	18/04/2024 8:00AM
Jueves B	25/04/2024 8:00AM
Viernes B	26/04/2024 8:00AM

1.1. EVALUACIÓN

- En la calificación se tendrán en cuenta los siguientes aspectos: documentación, diseño e implementación, diseño de casos de prueba, análisis de las pruebas realizadas y facilidad para la repetición de las pruebas por los profesores.
- Se aplicarán las reglas de tratamiento de casos de plagio explicadas en la presentación de la asignatura.

La valoración máxima de la práctica es de **10 puntos**.

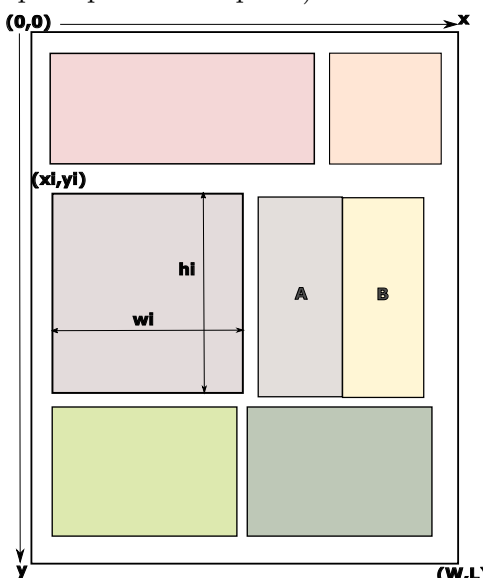
2. ENUNCIADO

El diseño de la página es muy importante para los periódicos y revistas. Los artículos se colocan en diferentes posiciones para maximizar el impacto visual. Cada artículo tiene un tamaño determinado y una ubicación preferencial fija. Cada artículo siempre cabe en una página, sin embargo no se pueden incluir todos en una sola página (tampoco pueden solaparse).

Podemos representar la página con un rectángulo, definido por las esquinas superior izquierda $(0,0)$ y inferior derecha (W,L) . Sean a_1, \dots, a_n los artículos a publicar: cada artículo a_i tiene que colocarse en un rectángulo (w_i, h_i, x_i, y_i) —donde w_i es la anchura, h_i la altura, y x_i, y_i las coordenadas cartesianas de la posición de la esquina superior izquierda.

Se quiere encontrar el conjunto de artículos a colocar en la página que maximice la cantidad de espacio ocupada (es decir, que maximice el área total ocupada por los artículos elegidos).

Obviamente los artículos no deben solaparse! Dos artículos se solapan si la intersección de los dos rectángulos correspondientes no es vacía: los artículos pueden compartir un lado, como por ejemplo los artículos A y B en la figura, o una esquina.



TAREA 1. DISEÑO Se pide diseñar un algoritmo **de búsqueda con retroceso** que, dados:

- La dimensión de la página (anchura W y altura L); y
- Una lista de n artículos, cada uno con su dimensión y posición (es decir, anchura w_i , altura h_i y coordenadas cartesianas (x_i, y_i) como especificado anteriormente),

determine los artículos a colocar en la página y calcule la cantidad de espacio total ocupada por los mismos.

TAREA 2. IMPLEMENTACIÓN Se pide desarrollar un programa que implemente el algoritmo de solución propuesto. La forma de ejecutar el programa será, por ejemplo, la siguiente:

```
> busca pruebas.txt resultados.txt
```

donde `busca` es el programa ejecutable, `pruebas.txt` es un fichero de texto que incluye los datos de diferentes *instancias* del problema y `resultados.txt` es un fichero de texto que guarda los resultados correspondientes. Los formatos del fichero de entrada y de salida se detallan a continuación.

Formato del fichero de entrada. El fichero de entrada está organizado en bloques: cada bloque es una instancia diferente del problema. La primera línea de cada bloque tiene tres números: el número de artículos n , la anchura W y altura H de la página. Las n líneas siguientes representan los artículos. Para cada artículo hay cuatro números: anchura w_i , altura h_i , y coordenadas cartesianas x_i, y_i . A continuación se muestra un ejemplo de fichero de entrada que especifica dos instancias del problema.

```
5 280 400
10 10 0 0
10 10 15 15
10 10 10 10
20 10 20 20
20 10 25 15
6 280 400
10 20 30 40
50 60 70 80
20 30 40 50
90 80 70 60
80 70 60 50
50 50 40 40
```

La dimensión de la página es la misma en las dos instancias ($280 \times 400 \text{ mm}$), en la primera hay $n = 5$ artículos a colocar mientras en la segunda hay $n = 6$ (también para los artículos se consideran las correspondientes medidas en mm).

Formato del fichero de salida. El fichero de salida tiene un número de líneas igual al número de bloques del fichero de entrada. Cada línea tiene que incluir como mínimo: el área total ocupada por los artículos elegidos y el tiempo de ejecución. Es opcional incluir los artículos a colocar en la página. A continuación se muestra un ejemplo de fichero de salida con la solución de las dos instancias especificadas en el fichero de entrada anterior (área total en mm y el tiempo en milisegundos).

```
400 0.209323
8000 0.355868
```

TAREA 3. EXPERIMENTACIÓN Analizar la corrección y eficiencia (tiempo de ejecución y número de nodos del árbol de búsqueda generados) del algoritmo implementado a través de un conjunto de pruebas.

2.1. BIBLIOGRAFÍA → EN MOODLE

1. Transparencias de la asignatura.
2. Las páginas web *Bibliografía de referencia* en los apartados *Búsqueda con retroceso*.