



Práctica 4: Ramificación y poda, programación lineal

1. CONSIDERACIONES GENERALES

- La entrega de la práctica se realizará en *Moodle* a través de la tarea *Entrega práctica 4*
- Se entregará un fichero `practica4.zip` que contiene un directorio denominado `practica4_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 `practica4_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 4 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 | 23/05/2024 8:00AM |
| Viernes A | 24/05/2024 8:00AM |
| Jueves B | 30/05/2024 8:00AM |
| Viernes B | 31/05/2024 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** así repartidos:

- Tareas 1,2 y 3: **8 puntos**
- Bola extra: **2 puntos**

Para esta última práctica no se organizarán sesiones de evaluación.

2. ENUNCIADO

Considera el siguiente problema “equivalente” al problema planteado en las prácticas 2 y 3:


Dados n artículos, encontrar el subconjunto que minimice la cantidad de espacio libre en la página.


Obviamente, se debe seguir cumpliendo la condición de no solape de los artículos a colocar en la página.

TAREA 1. DISEÑO Se pide diseñar un algoritmo de ramificación y poda de mínimo coste para resolver el problema.

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:

```
> buscaRyP pruebas.txt resultados.txt
```

donde `buscaRyP` 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 definen en el anexo A).* 

 **TAREA 3. EXPERIMENTACIÓN** Se pide analizar experimentalmente la corrección y eficiencia del algoritmo implementado a través de un conjunto de pruebas. En particular, con respecto a la eficiencia, utilizar el mismo conjunto de pruebas y comparar los resultados con los obtenidos de:

- La solución de la práctica 2 (tiempo de ejecución y número de nodos del árbol de búsqueda generados)
- Las soluciones de la práctica 3 (tiempo de ejecución)

Incluir consideraciones sobre el análisis de los resultados en el informe.

BOLA EXTRA. Esta tarea es **opcional** y consiste en:

1. Formalizar el problema general (con parámetros n, W, H, \dots) en un problema de programación lineal (entera) paramétrico.
2. Implementar un programa utilizando las API proporcionadas por una herramientas de programación lineal de libre elección para:

- a) Especificar el problema de programación lineal formalizado al punto 1.
- b) Resolver el problema de programación lineal con un *solver* de la herramienta elegida.

La forma de ejecutar el programa será, por ejemplo, la siguiente:

```
> buscaLP pruebas.txt resultados.txt
```

donde `buscaLP` 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. Véase los formatos de los dos ficheros en el anexo A).

3. Utilizar el mismo conjunto de pruebas de los experimentos en la Tarea 3 para:
 - a) Comprobar la corrección de la solución.
 - b) Comparar los tiempos de ejecución de las diferentes soluciones (backtracking, programación dinámica, ramificación y poda, y programación lineal).

2.1. REFERENCIAS

- Transparencias de la asignatura.
- La página en Moodle *Bibliografía de referencia* en el apartado *Ramificación y poda*.

Para la bola extra

- La página en Moodle *Bibliografía de referencia* en el apartado *Programación lineal y reducciones*.
- [Enlace interesante sobre restricciones de no solape de rectángulos](#).
- La página web en Moodle *Herramientas de programación lineal* en el apartado *Programación lineal y reducciones*.

A. ANEXO

A.1. 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).

A.2. 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:

1. La cantidad de espacio libre en la página y
2. El tiempo de ejecución.

Es opcional incluir los artículos a colocar en la página. <-- (Añadir esto al igual que en las otras prácticas)