



Práctica 3: Búsqueda con retroceso

1. CONSIDERACIONES GENERALES

- La entrega de la práctica se realiza en *Moodle* (tarea *Entrega práctica 3*).
- Hay que entregar un fichero comprimido `practica3.tar` (o `.zip`, `.gz`, etc.) de un directorio `practica3`.
- El directorio `practica3` tiene que incluir:
 1. Un fichero de texto `LEEME.txt` con la descripción general del directorio: cómo está organizado, instrucciones de instalación, compilación y ejecución, instrucciones para repetir las pruebas.
 2. Los ficheros fuente del código debidamente comentados.
 3. Un *shellscript* denominado `ejecutar.sh` que automatice la compilación y ejecución de los programas entregados con los casos de prueba.
 4. Los ficheros auxiliares de entrada necesarios para ejecutar las pruebas del punto anterior.
 5. Un informe con la descripción del diseño e implementación de la solución, de las pruebas y análisis de resultados (fichero PDF, máximo 4 páginas sin portada).
- **Fechas límite de entrega para la primera convocatoria:**

Grupo	Fecha y hora
Jueves A	27/03/2025 8:00AM
Viernes A	28/03/2025 8:00AM
Jueves B	03/04/2025 8:00AM
Viernes B	10/04/2025 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.

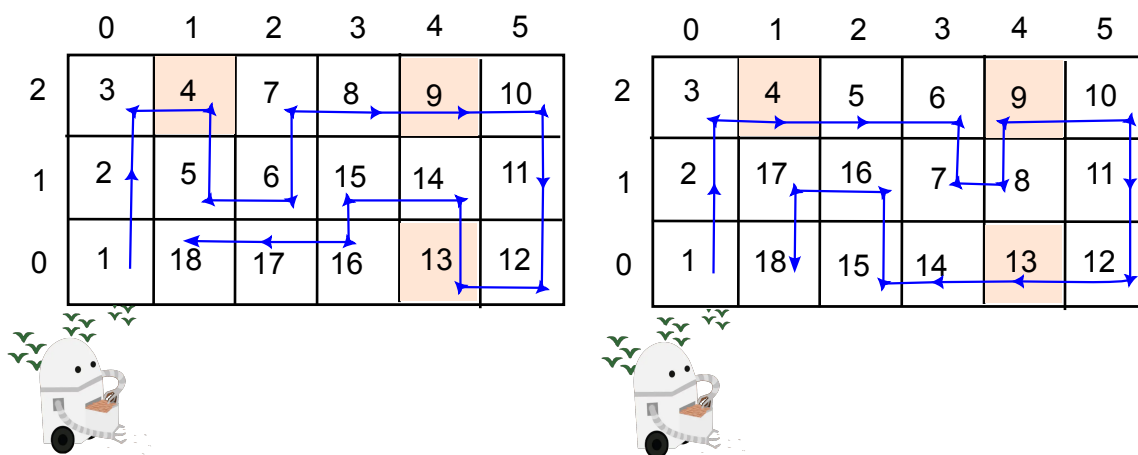
La valoración máxima de la práctica es de **10 puntos** (= 9 puntos de las tareas obligatorias + 1 punto de la bola extra).

2. ENUNCIADO

YuMi es un robot de la empresa *ABB* empleado en un proyecto como asistente para reforestar la Amazonia. YuMi automatiza las tareas de siembra de semillas como cavar un hoyo en el suelo, dejar caer la semilla, compactar el suelo en la parte superior y marcarlo con una etiqueta codificada, permitiendo así replantar un área del tamaño de dos campos de fútbol en un día en las zonas que requieren reforestación y además monitorizar el estado de la zona reforestada en los días siguientes. Los ingenieros de ABB controlan YuMi en tiempo real y de forma remota desde Suecia. Para facilitar la reforestación y el control de una zona, definen la zona como una cuadrícula rectangular y programan YuMi para que se desplace de cuadrado en cuadrado para realizar las tareas de siembra y control en cada cuadrado de la cuadrícula. Además especifican tres ubicaciones en la cuadrícula como puntos de registro: YuMi envía un informe del estado desde estos puntos cuando haya completado un cuarto, la mitad y tres cuartos de su recorrido. YuMi debe comenzar su recorrido por la cuadrícula desde la esquina inferior izquierda, designada en las coordenadas (fila, columna) como $(0,0)$, visitando cada uno de los demás cuadrados de la cuadrícula exactamente una vez y finalizando su recorrido en cuadrado $(0,1)$. Dada una zona a reforestar y monitorizar, si hay varios recorridos que YuMi puede seguir, se debe utilizar uno diferente cada día. El robot puede moverse de un cuadrado a uno adyacente, en cada paso (se puede mover en las cuatro direcciones: norte, sur, este u oeste).

TAREA 1. DISEÑO Se pide diseñar un algoritmo **de búsqueda con retroceso** para calcular cuántos recorridos diferentes puede hacer YuMi en una zona a reforestar dado su tamaño y los tres puntos de registro.

Por ejemplo, supongamos una zona de tamaño 3×6 con los puntos de registro, en orden de visita, $(2,1)$, $(2,4)$ y $(0,4)$. Yumi tiene que empezar en $(0,0)$ y finalizar en $(0,1)$ después de visitar los 18 cuadrados. Debe visitar el cuadrado $(2,1)$ en el paso $4 (= \lfloor 18/4 \rfloor)$, el cuadrado $(2,4)$ en el paso $9 (= \lfloor 18/2 \rfloor)$ y el cuadrado $(0,4)$ en el paso $13 (= \lfloor 3 \cdot 18/4 \rfloor)$. En el ejemplo, hay únicamente dos posibles recorridos (representados en la figura). Observar que cuando el tamaño de la zona no es divisible por 4 se usa la división entera para determinar los pasos de control.



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:

```
> recorridosYumi pruebas.txt resultados.txt
```

donde `recorridoYumi` es el programa ejecutable, `pruebas.txt` es un fichero de texto que incluye los datos de diferentes casos de prueba y `resultados.txt` es un fichero de texto que guarda los resultados de los casos de prueba. 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 un caso de prueba. La primera línea de cada bloque incluye dos números enteros que especifican el tamaño $m \times n$ de la cuadrícula: m número de filas, n número de columnas. La siguiente línea contiene seis números enteros que especifican las coordenadas (r_i, c_i) , $0 < r_i < m$, $0 < c_i < n$ de los tres puntos de control $i = 1, 2, 3$.

A continuación se muestra un ejemplo de fichero de entrada que especifica cuatro casos de prueba, con diferentes dimensiones de la cuadrícula. En particular, el segundo caso de prueba corresponde al ejemplo considerado anteriormente.

```
2 3
0 0 1 1 1 2
3 6
2 1 2 4 0 4
4 3
2 0 3 2 0 2
6 6
3 1 5 4 2 4
```

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 corresponde a la solución del bloque correspondiente en el fichero de entrada y tiene que incluir:

1. El número total de recorridos diferentes que empiezan en $(0, 0)$, terminan en $(0, 1)$ y visitan los tres puntos de control (r_i, c_i) al paso $\lfloor i \cdot m \cdot n / 4 \rfloor$, para $i = 1, 2, 3$.
2. El tiempo de ejecución.

A continuación se muestra un ejemplo de fichero de salida con la solución de los casos de prueba especificados en el fichero de entrada anterior (número de recorridos y el tiempo en milisegundos)¹.

```
1 0.026942
2 0.226205
0 0.040153
24 6.27125
```

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. En la experimentación se deberían considerar como mínimo los casos de pruebas proporcionados en Moodle (fichero `pruebas.txt`).

BOLA EXTRA. Proponer un nuevo algoritmo de búsqueda con retroceso aplicando el enfoque *meet in the middle*. Comparar los tiempos de ejecución de esta nueva solución con los tiempos de ejecución de la solución propuesta anterior.

2.1. BIBLIOGRAFÍA → EN MOODLE

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

¹El tiempo de ejecución es orientativo, depende de la implementación y del entorno de ejecución.