

Práctica 6 – Complejidad de los problemas combinatorios (Cuadrados latinos)

Definición del problema y objetivo

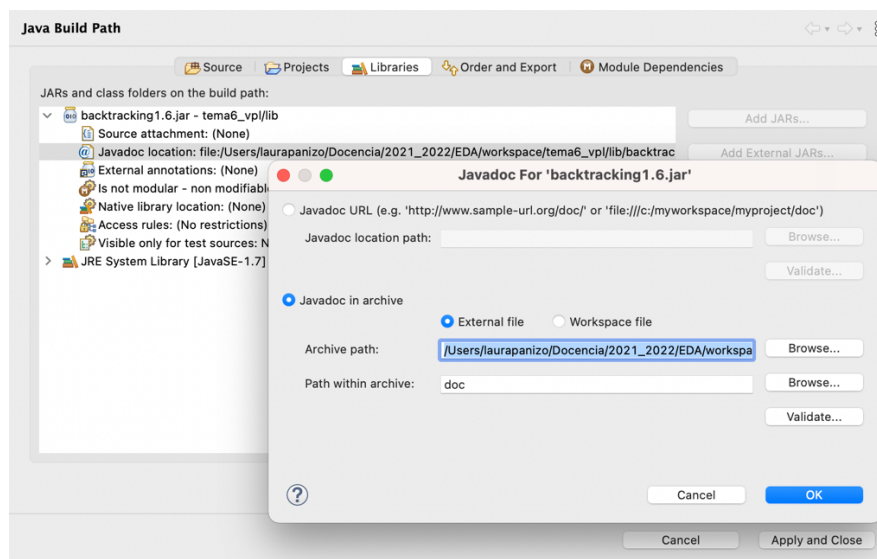
El objetivo de esta práctica es estudiar la complejidad de la resolución de problemas combinatorios que presentan cambio de fase (phase transition). Para ello se va a implementar un algoritmo de backtracking que resuelve el problema de los cuadrados latinos. En el documento [latinsquares.pdf](#) (disponible en el campus virtual) puedes encontrar la definición del problema y los algoritmos.

Realización de la práctica

El estudiante tiene que implementar los métodos `initialState` y `backtracking` de la clase `LatinSquareBacktracking`.

Además de la clase `LatinSquareBacktracking`, se proporcionan las siguientes clases y librerías:

- `backtracking1.6.jar` y `backtracking1.6-doc.jar`: esta librería contiene algunas clases necesarias para la implementación de los métodos (por ejemplo, la clase `Pair`) y correcta ejecución del programa.
- Para importar la librería en el proyecto eclipse:
 - Copiar los ficheros jar en el proyecto. Por ejemplo, puedes crear una carpeta `lib` dentro del proyecto para mantener los ficheros organizados.
 - Sobre el nombre del proyecto, pulsa el botón secundario y selecciona `Configure Build Path > Pestaña Libraries`
 - Si en el área central te aparecen `Module Path` y `Class Path` selecciona `Class Path`
 - Pulsa el botón `Add JARs` y selecciona el fichero `rearrangement1.6.jar`
 - Para añadir la documentación asociada despliega la librería asociada al jar, y haz doble click sobre `Javadoc location`. Selecciona `Javadoc in archive > External file`. En el path añade el fichero `backtracking1.6-doc.jar`, y en `Path within archive` selecciona `doc` (como aparece en la imagen).



- `TestLatinSquareBacktracking`: en esta clase se encuentra en método `main` (no hay que modificarlo). El método `main` recibe en los argumentos el nombre del fichero que contiene el cuadrado latino que se quiere resolver. Como resultado se muestra por consola el tiempo se ha tardado en resolver el cuadrado latino y, además, se generan dos ficheros: uno con la solución al cuadrado latino (`solution_<nombre_fichero_entrada>`) y otro con

- estadísticas (stats_<nombre_fichero_entrada>).
- Fichero con cuadrados latino de prueba (ls5a.txt, ls5b.txt) y la correspondiente solución (solution_ls5a.txt, solution_ls5b.txt).
 - Fichero con cuadrados latinos para realizar el análisis de la complejidad (ls10.txt)
 - analyze.R: este fichero contiene un script que procesa el fichero solution_ls10.txt. Hay una guía en el campus que explica como instalar el entorno R y ejecutar el script.

Entrega y evaluación

La evaluación de la práctica consiste en la entrega y evaluación automática del código del alumno y una memoria (de no más de 3 páginas) en la que el alumno presenta los objetivos, el entorno de ejecución donde se ha realizado las pruebas, los resultados teóricos esperados y los resultados que ha obtenido en el estudio experimental de la complejidad del algoritmo. A continuación, se detalla la nota que se puede obtener:

Nota	Descripción
0	No pasa el test en laboratorio ni entrega memoria o el sistema detecta plagio en el código o la memoria
[1,4]	No pasa el test en laboratorio. Entrega la memoria y pasa los test fuera de plazo
[4,6]	Pasa parcialmente los test de laboratorio. Entrega memoria y pasa test fuera de plazo
[6,10]	Pasa los test en laboratorio y entrega memoria

Para la evaluación automática del código el estudiante debe subir el fichero LatinSquareBacktracking.java a una actividad del campus que realizará una serie de pruebas sobre la implementación. El código hay que subirlo siempre al final de la sesión de laboratorio correspondiente, y si no se pasan todas las pruebas tendrá una segunda oportunidad fuera de plazo. La fecha se comunicará al final de la sesión de laboratorio.