

Práctica 1. Backtracking

- Solitario en cruz

El objetivo principal de esta práctica es la resolución de un “solitario en cruz” mediante el esquema de backtracking.

El “solitario en cruz” es un rompecabezas en el que se dispone de un tablero con 33 posiciones o celdas en forma de cruz. Inicialmente todas las posiciones están ocupadas por canicas excepto la posición central que permanece vacía.

Las reglas del este juego son simples:

- Se puede tomar una canica y saltar sobre otra canica vecina siempre que haya un espacio vacío adyacente a la canica saltada, que se retirará del tablero.
- El rompecabezas se resuelve si al final sólo existe una canica y se encuentra en el centro del tablero.

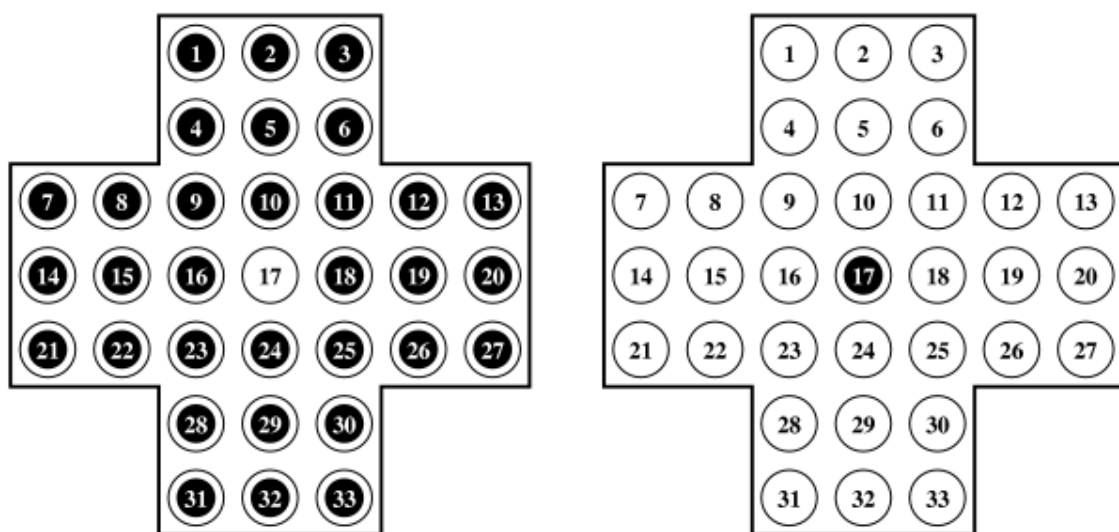


Figura 1. Solitario en cruz. El tablero está formado por 33 celdas posicionadas formando una cruz. *Izquierda:* situación inicial del tablero en el que hay 32 canicas y un hueco (en la posición central del tablero). *Derecha:* situación objetivo del tablero en el que sólo queda una canica posicionada en la celda central del tablero (la que en el tablero inicial tenía un hueco).

Al principio del juego sólo hay cuatro movimientos válidos (canica 5 salta a canica 10 y se posiciona en hueco 17; canica 15 salta a canica 16 y se posiciona en hueco 17; canica 29 salta a canica 24 y se posiciona en hueco 17; canica 19 salta a canica 18 y se posiciona en hueco 17), pero las opciones aumentan durante el juego ya que se van generando más huecos en el tablero (la canica sobre la que se salta al hacer un movimiento se retira del tablero). La Figura 2 muestra un ejemplo de la evolución de este rompecabezas.

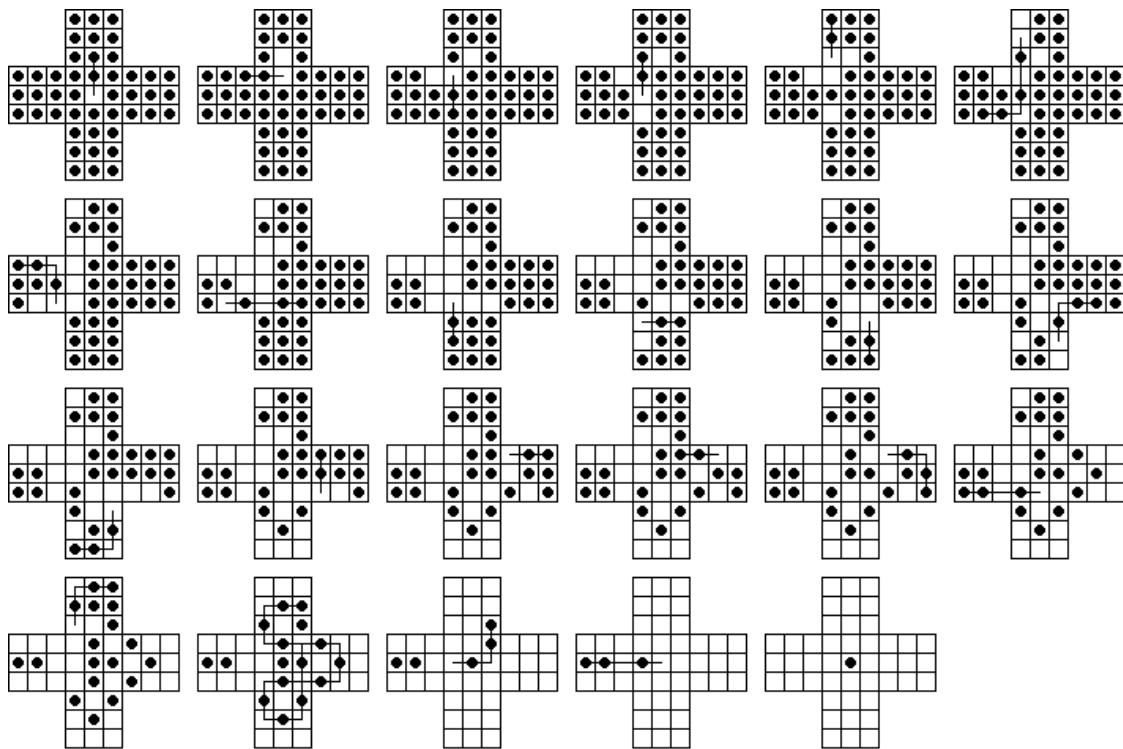


Figura 2. Ejemplo de evolución de un tablero de 45 celdas, desde la situación inicial hasta la situación final. Téngase en cuenta que alguno de los tableros de la figura muestra más de un movimiento (por ejemplo, el tablero de la esquina superior derecha muestra tres movimientos consecutivos)

SE PIDE:

Implementar un algoritmo de backtracking recursivo en Java, que encuentre una solución a este rompecabezas. La solución del rompecabezas deberá devolverse como una secuencia de movimientos.

Para desarrollar esta práctica se suministran 5 clases Java:

- **Booleano**. La clase *Booleano* envuelve un valor del tipo primitivo *boolean* en un objeto. Un objeto de tipo *Booleano* contiene un único campo cuyo tipo es *boolean*. Esta clase se suministra completamente implementada (no deberá modificarse el código de la misma).
- **Posicion**. La clase *Posicion* representa una posición concreta en un tablero organizado en filas y columnas. Esta clase se suministra completamente implementada (no deberá modificarse el código de la misma).
- **Movimiento**. La clase *Movimiento* representa un movimiento del solitario en cruz, mantenido éste como la celda origen, la celda que se salta y la celda destino. Esta clase se suministra completamente implementada (no deberá modificarse el código de la misma).
- **Tablero**. Esta clase representa el tablero de juego, así como la funcionalidad necesaria para manejarlo. Esta clase se suministra parcialmente implementada (deberán implementarse todos los métodos incluidos en la

misma, y en ningún caso deberán añadirse/modificarse/eliminarse los métodos públicos cuya cabecera se suministra en esta clase).

- ***Solitario***. *Esta es la clase principal (aplicación) del juego del solitario en cruz resuelto mediante el esquema de backtracking*. Esta clase se suministra parcialmente implementada. Concretamente se deberá incluir el código del método que busca la solución del solitario en cruz mediante el esquema de backtracking.

El código de estas cinco clases se suministra documentado con el fin de aclarar la funcionalidad que suministra cada método (en el caso de los métodos implementados) o debe suministrar (en el caso de los métodos cuyo contenido debe implementarse).

ENTREGA DE LA PRÁCTICA

La entrega consistirá en subir a la plataforma moodle un archivo en formato comprimido (ZIP o RAR) que incluya las clases **Tablero.java** y **Solitario.java** (sólo los ficheros .java) completamente implementadas. Si la práctica se ha desarrollado individual, el nombre del fichero comprimido deberá ser el número de matrícula del alumno (ejemplo: ai0260.zip). Si la práctica se ha desarrollado en pareja con otro compañero, el nombre del fichero comprimido deberá ser el número de matrícula de ambos (ejemplo: ai0340bc0433.zip). En este último caso, **AMBOS** alumnos deberán subir la práctica al moodle.

El archivo comprimido se deberá subir a la plataforma moodle antes del 3 de noviembre de 2015 a las 23:55. Aquellos alumnos que hayan subido la práctica deberán realizar un examen escrito de la misma, la semana del 2 de noviembre de 2015 en su horario de prácticas (se avisará a través del moodle del lugar exacto de realización del examen). Para poder aprobar esta práctica será indispensable haberla subido a la plataforma, haber superado el examen de la misma y haberla desarrollado correctamente.