

PROGRAMACIÓN Y ESTRUCTURAS DE DATOS AVANZADAS

Febrero 2017 (Primera semana)

Normas de valoración del examen:

- La nota del examen representa el 80% de la valoración final de la asignatura (el 20% restante corresponde a las prácticas).
- Cada cuestión contestada correctamente vale 1 punto.
- Cada cuestión contestada incorrectamente baja la nota en 0.3 puntos.
- Debe obtenerse un mínimo de 3 puntos en las cuestiones para que el problema sea valorado (con 3 cuestiones correctas y alguna incorrecta el examen está suspenso).
- La nota total del examen debe ser al menos de 4.5 para aprobar.
- Las cuestiones se responden en una hoja de lectura óptica.

SOLUCIONES:

Test:

Tipo A: 1B 2C 3A 4C 5C 6D

Tipo B: 1C 2D 3C 4C 5A 6B

Problema (4 puntos).

Se distribuyen 32 fichas (*) sobre un tablero como muestra la figura:

```
      # # #
    # * * * #
  # * * * #
# * * * * * * * #
# * * *   * * * #
# * * * * * * * #
    # * * * #
    # * * * #
      # # #
```

El juego consiste en mover las fichas hasta que solo quede una, con las siguientes condiciones:

- Las fichas * se mueven saltando una sobre otra, desapareciendo la ficha sobre la que se salta. No se salta en diagonal. Las # indican fin del tablero. No se puede saltar sobre ellas.
- Solo se salta sobre una ficha (y solo una) si tras ella hay un hueco.

Se pide programar un algoritmo que resuelva el problema en el menor número de movimientos.

La resolución de este problema debe incluir, por este orden:

1. Elección del esquema más apropiado, el esquema general y explicación de su aplicación al problema (0,5 puntos).
2. Descripción de las estructuras de datos necesarias (0,5 puntos solo si el punto 1 es correcto).
3. Algoritmo completo a partir del refinamiento del esquema general (2,5 puntos solo si el punto 1 es correcto). Si se trata del esquema voraz, debe realizarse la demostración de optimalidad. Si se trata del esquema de programación dinámica, deben proporcionarse las ecuaciones de recurrencia.
4. Estudio del coste del algoritmo desarrollado (0,5 puntos solo si el punto 1 es correcto).

Solución:

1. El esquema más apropiado es vuelta atrás. Hay que destacar que el algoritmo no precisa estimar el coste de la solución, sino el número de movimientos. No tiene sentido el esquema de ramificación y poda porque la estimación del número de movimientos hasta la solución es conocida (31) y además no puede variar de una solución a otra.
2. Se puede representar el tablero como una matriz bidimensional en la que los elementos pueden tomar los valores *novalida*, *libre* u *ocupada*, dependiendo de que esa posición no sea válida (no corresponda a una de las posición que forman la “cruz”), o bien siendo válida exista ficha o no.

La solución vendrá dada en una matriz $X = [x_1, x_2, \dots, x_m]$ en donde x_i representa un movimiento (salto) del juego. Cada movimiento se representará con la posición de la ficha que va a efectuar el movimiento, la posición hacia donde da el salto y la posición de la ficha “comida”. El valor m representa el número de movimientos (saltos) efectuados para alcanzar la solución. Puesto que en cada movimiento ha de comerse obligatoriamente una ficha, sabemos que m podrá tomar como máximo valor 31.

3. Para más detalles consultar: R. Guerequeta, A. Vallecillo (1999). Técnicas de Diseño de Algoritmos Ed. Univ. Málaga, pág. 245.
4. El coste de la solución sería: 4^{31}