Practica 1. Resolución de problemas con búsqueda. Parte IV.

Se pide resolver **alguno de los siguientes problemas que se plantean**. Para cada problema se debe proponer una representación adecuada que permita a un agente **resolverlo con distintos parámetros de entrada** (tamaños, valores ,..). Para cada problema hay que entregar una breve memoria (usar formato ipynb) y justificar cuál es el/los algoritmo/s de búsqueda utilizados para resolverlo incluyendo datos de la complejidad de la resolución del problema (tiempos aproximados y comportamiento en cuanto a optimalidad de las soluciones encontradas).

El objetivo de la práctica es usar la librería AIMA y proporcionar una solución completa al problema con ejecución y pruebas. Como alternativa se puede optar por una resolución teórica de los problemas entregando la memoria de la representación del problema (estado inicial, final, especificación de los operadores y heurística).

Fecha de entrega de la parte IV: **9 de abril**

Problema 1. (resolverlo permitiendo cambiar el numero de personas y los tiempos)

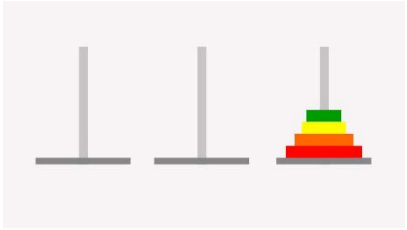
Un grupo de 5 personas quiere cruzar un viejo y estrecho puente. Es una noche cerrada y se necesita llevar una linterna para cruzar. El grupo solo dispone de una linterna, a la que le quedan 5 minutos de batería.

1. Cada persona tarda en cruzar 10, 30, 60, 80 y 120 segundos, respectivamente.
2. El puente solo resiste un máximo de 2 personas cruzando a la vez, y cuando cruzan dos personas juntas, caminan a la velocidad del más lento.
3. No se puede lanzar la linterna de un extremo a otro del puente, así que cada vez que crucen dos personas, alguien tiene que volver a cruzar hacia atrás con la linterna a buscar a los compañeros que falten, y así hasta que hayan cruzado todos.

¿Qué longitud tiene la solución del problema para cruzar a 5 personas?

¿Se puede cruzar el puente con una linterna de 15 minutos de batería si hay 10 personas que tardan tiempos 10, 20, 30, 40, 50, 60, 70, 80, 120 y 120? ¿Cuál es la solución obtenida? ¿Puedes usar los algoritmos vistos para calcular la batería mínima que necesito para cruzar a las 10 personas?

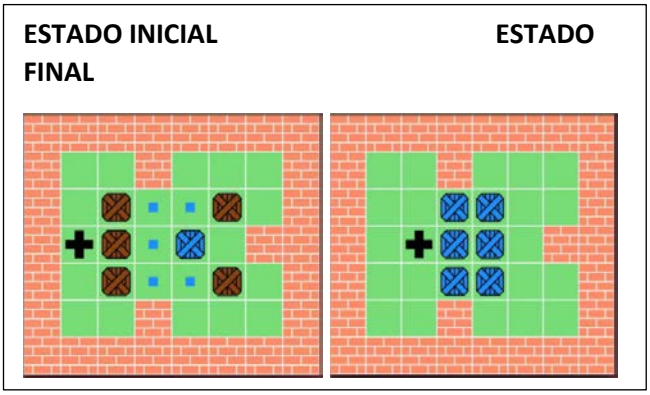
Problema 2. Torres de Hanoi:
Se tienen N discos de distinto tamaño apilados sobre una base A de manera que cada disco se encuentra sobre uno de mayor radio. Existen otras dos bases vacías B y C.
Haciendo uso únicamente de las 3 bases, el objetivo es llevar todos los discos de la base A hasta la base C.
Sólo se puede mover un disco a la vez, y cada disco puede descansar solamente en las bases o sobre otro disco de tamaño superior, pero no en el suelo.



¿Cuál es el factor de ramificación r del problema? ¿Qué algoritmo usarías para calcular la solución para 5 discos? Nota: Se sabe que el número mínimo de movimientos necesarios para resolver un rompecabezas de la Torre de Hanoi es 2n - 1, donde n es la cantidad de discos, es decir para n=5 hay una solución de longitud 31 y para n=6 de longitud 63

¿Cuánto tiempo ha tardado el algoritmo? ¿cómo se puede mejorar? Si no lo has programado puedes hacer el cálculo teórico usando el factor de ramificación y la profundidad de la solución.

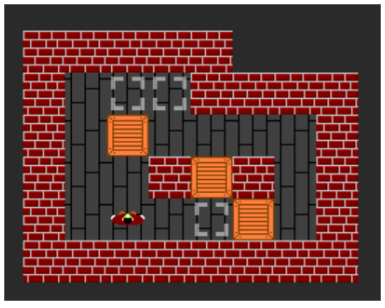
Problema 3. Se quiere desarrollar un agente inteligente para jugar al clásico juego japonés del Sokoban en un tablero NxM. En este juego, el agente puede empujar unos bloques (o cajas) por el tablero de uno en uno. En el tablero existen casillas vacías (transitables), muros (no transitables) y unas casillas especiales o metas. Hay tantas cajas como metas y el juego termina cuando todas las cajas están colocadas sobre las metas.
Las reglas del juego son las siguientes:



- El jugador puede moverse sobre casillas vacías o metas avanzando una casilla en cada movimiento.
- El jugador puede empujar cajas en la dirección de su movimiento, pero no tirar de ellas.
- El jugador sólo puede moverse en horizontal y vertical, no en diagonal.
- El jugador no puede atravesar cajas ni situarse sobre ellas.
- Las casillas tipo muro no pueden ser atravesados ni por el jugador ni por cajas. Puede haber casillas muro en cualquier posición del tablero menos en las casillas metas.
- Las cajas pueden encontrarse sobre casillas vacías o sobre metas.
- El jugador no puede empujar más de una caja a la vez. Si hay dos cajas juntas no se pueden empujar.
- Se puede suponer que los tableros iniciales son válidos, es decir, la configuración inicial del tablero tiene solución.

No hay que hacer interfaz para el juego. Existen muchísimas versiones para todas las plataformas. Podéis buscar versiones para jugar y comprender mejor el juego.
http://www.rodoval.com/heureka/sokoban/sokoban.html
Aunque no es necesario utilizarlo existe un formato de texto estándar para cargar niveles (XSB)

```
#####
# ..####
# $    #
#  #$# #
# @ .$ #
########
```



El significado de los distintos caracteres es el siguiente:

```
# -> pared    $ -> caja     . -> meta    * -> caja en meta
@ -> jugador  + -> jugador en meta
```