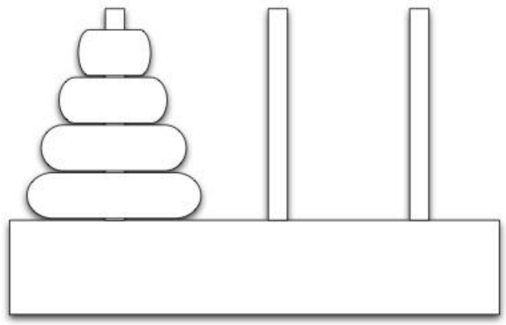
1. Torres de Hanoi



Las Torres de Hanoi es un famoso problema para estudiar la recursividad en informática y las ecuaciones de recurrencia en matemática discreta. Empezamos con N discos de distintos tamaños en un poste (apilados en orden según su tamaño) y dos postes vacíos. Se nos permite mover un disco de un poste a otro, pero nunca se nos permite mover un disco más grande encima de un disco más pequeño. El objetivo es mover todos los discos hasta el poste situado más a la derecha (véase la figura anterior).

En este problema, formularemos las Torres de Hanoi como un problema de búsqueda.

(a) Proponer una representación de estados para el problema.

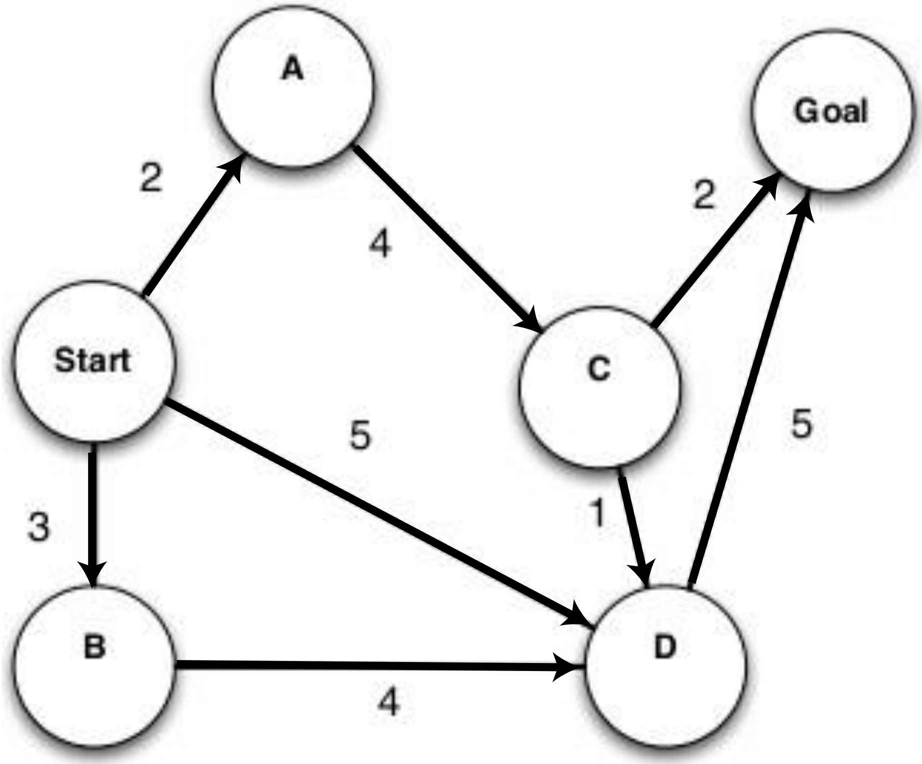
(b) ¿Cuál es el tamaño del espacio de estados?

(c) ¿Cuál es el estado inicial?

(d) Desde cualquiera de los estados, ¿qué acciones son legales?

(e) ¿Cuál es el test objetivo?

2. Algoritmos de búsqueda en acción



Para cada una de las siguientes estrategias de **búsqueda en grafos**, calcule el orden en que se expanden los estados, así como la ruta devuelta por la búsqueda en grafos. En todos los casos, supongamos que los lazos se resuelven de tal manera que los estados con orden alfabético anterior se expanden primero. Recuerde que, en la búsqueda en grafos, un estado se expande solo una vez.

(a) Búsqueda primero en profundidad

(b) Búsqueda primero en anchura

(c) Búsqueda de coste uniforme

3. El lobo, la oveja y la col

Considere el siguiente problema:

> *Un hombre se encuentra en la orilla izquierda de un río junto con un lobo, una oveja y una col. Quiere cruzar el río llevando consigo el lobo, la oveja y la col. En la barca sólo hay dos plazas, una de las cuales debe ir ocupada por el hombre. Cada uno de los restantes pasajeros (lobo, oveja, col) ocupa una plaza, de tal modo que sólo uno puede acompañar al hombre en cada viaje. Además, no puede dejar solos en una orilla al lobo con la oveja, ni a la oveja con la col (ni los tres), porque la primera se comería a la segunda en cada paso*

a) Proponga una representación de estados para el problema, e indique los estados inicial y objetivo

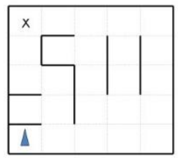
b) Simule la estrategia de búsqueda en anchura para este problema, asumiendo que se filtran todos los estados repetidos. Expanda los sucesores de los nodos siguiendo las preferencias del hombre, las cuales se ordenan (de mayor a menor) como sigue: viajar con la oveja, viajar con la col, viajar sólo, viajar con el lobo. Dibuje el árbol de búsqueda correspondiente e indique el orden en el que se exploran los nodos.

c) Simule ahora la estrategia de búsqueda en profundidad para este problema. Dibuje el árbol de búsqueda correspondiente e indique el orden en el que se exploran los nodos, asumiendo que se filtran todos los estados repetidos y siguiendo las preferencias del apartado anterior.

d) Suponga ahora que no se filtra ningún estado repetido. ¿La búsqueda en anchura y la búsqueda en profundidad seguirían siendo completas? Razone brevemente su respuesta

4. Búsqueda e heurísticas

Imagine que un agente similar a un automóvil desea salir de un laberinto como el que se muestra a continuación:



1. El agente es direccional y en todo momento mira hacia alguna dirección $d \in (N,S,E,O)$.
2. Con una sola acción, el agente puede avanzar a una velocidad ajustable *v o* girar. Las acciones de giro son *izquierda* y *derecha*, que cambian la dirección del agente en 90 grados. El giro solo está permitido cuando la velocidad es cero (y la deja en cero).
3. Las acciones de movimiento son *rápido* y *lento*. *Rápido* incrementa la velocidad en 1 y *lento* disminuye la velocidad en 1; en ambos casos, el agente mueve un número de cuadrados igual a su NUEVA velocidad ajustada (ver ejemplo a continuación).
4. Una consecuencia de esta formulación es que es **imposible** que el agente se mueva a la misma velocidad distinta de cero durante dos pasos de tiempo consecutivos.
5. Cualquier acción que resulte en una colisión con una pared o reduzca *v por* debajo de 0 / por encima de una velocidad máxima $V_{máx}$ es ilegal.
6. El objetivo del agente es encontrar un plan que lo aparque (estacionario) en la plaza de salida utilizando la menor cantidad posible de acciones (pasos de tiempo).

Como ejemplo: si en el paso de tiempo t la velocidad actual del agente es 2, al tomar la acción *rápido*, el agente primero aumenta la velocidad a 3 y se mueve 3 cuadrados hacia adelante, de modo que en el paso de tiempo t + 1 la velocidad actual del agente será 3 y estará a 3 casillas de distancia de donde estaba en el paso de tiempo t. Si, en cambio, el agente toma la acción *lento*, primero disminuye su velocidad a 1 y luego mueve 1 cuadrado hacia adelante, de modo que en el paso de tiempo t + 1 la velocidad actual del agente será 1 y estará a 1 casilla de distancia de donde estaba en el paso de tiempo t. Si, con una velocidad instantánea de 1 en el paso de tiempo t + 1, toma la acción *lento* nuevamente, la velocidad actual del agente se convertirá en 0 y no se moverá en el paso de tiempo t + 1. Luego, en el paso de tiempo t + 2, será libre de girar si lo desea. Tenga en cuenta que el agente no podría haber girado en el paso de tiempo t + 1 cuando tenía una velocidad de 1, porque tiene que estar estacionario para girar.

1. Si la cuadrícula *es M* por *N*, ¿cuál es el tamaño del espacio de estados? Justifique su respuesta. Debe suponer que todas las configuraciones son accesibles desde el estado inicial.
2. ¿Es admisible la distancia de Manhattan desde la ubicación del agente hasta la ubicación de la salida? ¿Por qué o por qué no?

3. Declarar y justificar una heurística admisible no trivial para este problema que no es la distancia de Manhattan a la salida.

4. Si usáramos una heurística inadmisible en la búsqueda en el grafo A*, ¿sería completa? ¿Sería óptima?

5. Si utilizamos una heurística *admisible* en la búsqueda en el grafo A*, ¿se garantiza que devolverá una solución óptima? ¿Y si la heurística fuera consistente? ¿Qué pasaría si estuviéramos usando la búsqueda de árbol A* en lugar de la búsqueda en el grafo A*?

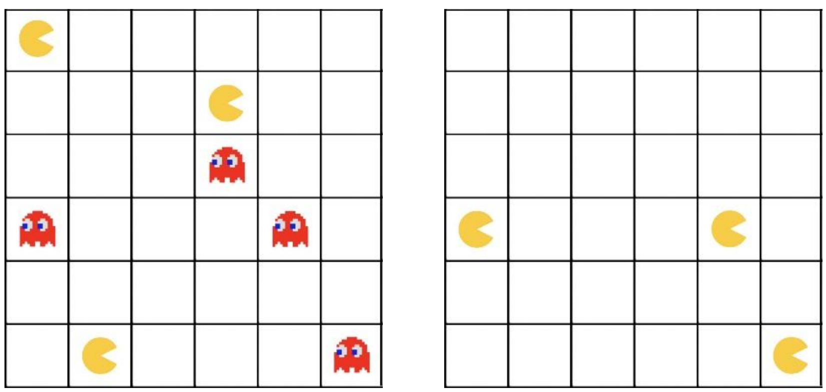
6. Dar una ventaja general que una heurística inadmisible podría tener sobre una admisible.

5. Pacamigos unidos

Pacman y sus Pacamigos han decidido combinar fuerzas e ir a la ofensiva, ¡y ahora están persiguiendo fantasmas! En una cuadrícula de tamaño M por N, Pacman y $P - 1$ de sus Pacamigos se están moviendo para eliminar colectivamente **a todos los** fantasmas en la cuadrícula pisando la misma casilla que cada uno de ellos. Pasar a la misma casilla que un fantasma lo eliminará de la cuadrícula y moverá al Pacman a esa casilla.

En cada turno, Pacman y sus Pacamigos pueden elegir una de las siguientes cuatro acciones: *izquierda, derecha, arriba, abajo,* pero no pueden chocar entre sí. En otras palabras, cualquier acción que resulte en dos o más Pacmen ocupando la misma casilla resultará en ningún movimiento ni para Pacman ni para los Pacamigos. Además, Pacman y sus Pacamigos son **indistinguibles** entre sí. Hay un total de G fantasmas, que son indistinguibles entre sí y no pueden moverse.

Tratando esto como un problema de búsqueda, consideramos que cada configuración de la cuadrícula es un estado, y el estado objetivo es la configuración donde **todos los** fantasmas han sido eliminados del tablero. A continuación, se muestra un ejemplo de estado inicial, así como un ejemplo de estado de objetivo:



(a) Posible estado inicial (b) Posible estado objetivo

Suponga que cada una de las siguientes subpartes es **independiente** entre sí. **También suponga que, independientemente de cuántos Pacmans se muevan en un turno, el coste total del movimiento sigue siendo 1.**

(a) Supongamos que Pacman no tiene Pacamigos, por lo que $P = 1$.

(i) ¿Cuál es el tamaño de la representación mínima del espacio de estados dada esta condición?

Para cada una de las siguientes heurísticas, seleccione si la heurística es sólo admisible, sólo consistente, ninguna o ambas. Recordemos que $P = 1$.

(ii) $h(n)$ = la suma de las distancias de Manhattan de Pacman a cada fantasma.

(iii) $h(n)$ = el número de fantasmas multiplicado por la distancia máxima de Manhattan entre Pacman y cualquiera de los fantasmas.

(iv) $h(n)$ = el número de fantasmas restantes.

(b) Supongamos que Pacman tiene exactamente un Pacamigo menos que el número de fantasmas; por lo tanto, $P = G$. Recordemos que Pacman y sus Pacamigos son indistinguibles entre sí.

(i) (Opcional) ¿Cuál es el tamaño de la representación mínima del espacio de estado dada esta condición? Recordemos que $P = G$.

Para cada una de las siguientes heurísticas, seleccione si la heurística es sólo admisible, sólo coherente, ninguna o ambas. Recordemos que $P = G$.

(ii) $h(n)$ = la mayor de las distancias de Manhattan entre cada Pacman y su fantasma más cercano.

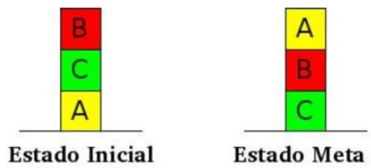
(iii) $h(n)$ = la más pequeña de las distancias de Manhattan entre cada Pacman y su fantasma más cercano.

(iv) $h(n)$ = número de fantasmas restantes

(v) $h(n)$ = número de fantasmas restantes / P

6. Problema de los bloques

Considere el problema de los bloques cuyo estado inicial y estado objetivo se muestran en la siguiente figura:



Las acciones posibles son:

- Situar un bloque libre (sin nada encima) en la mesa
- Situar un bloque libre (sin nada encima) sobre otro bloque libre

Desarrolle el árbol de búsqueda que expande el algoritmo A^*, utilizando la siguiente heurística:

$$h(n) = \textit{número de bloques descolocados}$$

Con tal fin, considere que un bloque está descolocado si *por debajo* no tiene el elemento correcto (bien el bloque deseado o bien la mesa). Utilizando la búsqueda en el grafo, indique el orden de expansión de los estados y muestre en cada paso los valores de f, g y h. Suponga que el coste de cada movimiento es 1.