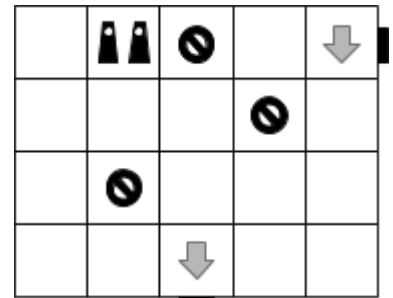


Búsqueda.

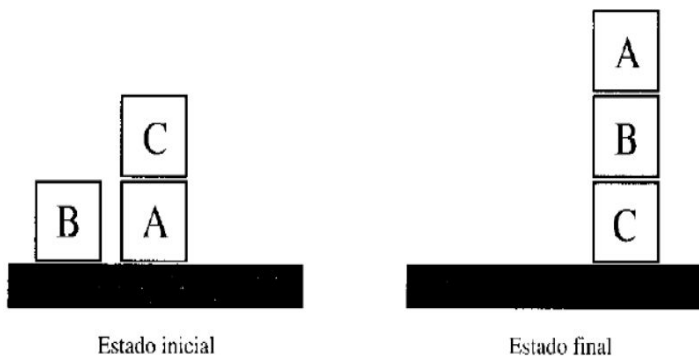
Robots defensivos secretos.

Se poseen dos robots para custodiar un centro de investigación secreto. Los robots permanecen almacenados en espera, y frente a alguna emergencia deben reaccionar ubicándose en posiciones defensivas. El centro de investigación posee habitaciones en forma de grilla, pero algunas de ellas no son accesibles para los robots, por riesgo de contaminación. Los robots sólo pueden moverse entre habitaciones adyacentes.



Se debe resolver la “puesta en defensiva” como un problema de búsqueda, con el objetivo de encontrar el camino más óptimo.

Anomalía de Sussman.

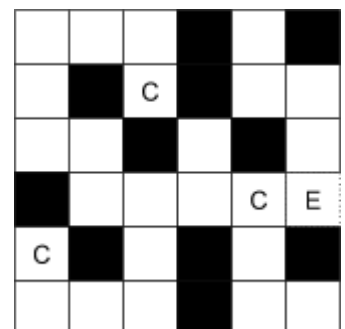


Se tiene un tablero con bloques sobre el mismo. Además se cuenta con un brazo robot capaz de tomar un bloque a la vez y moverlo. Se desea encontrar una secuencia de acciones para llegar al estado final, a partir del estado inicial. Se tiene como restricción que no se puede mover un bloque que tenga otro encima.

Ratabots

Luego de que ambientalistas logren prohibir el uso de animales vivos para experimentos, un grupo de científicos se encuentra con la necesidad de programar un robot que simule el comportamiento de ratas de laboratorio dentro de laberintos. El robot debe poder encontrar 3 “comidas” escondidas dentro, “ingerirlas”, y salir del laberinto al finalizar su “alimentación”.

El laberinto, con la entrada y las comidas, es el que se ve en la figura.



Jarros

Se tienen N jarros enumerados de 1 a N , donde la capacidad en litros del jarro I es I . Esto es, el jarro 1 tiene capacidad de 1 litro, el jarro 2, 2 litros y así sucesivamente. Inicialmente el jarro N está lleno de agua y los demás están vacíos.

El objetivo es que todos los jarros queden con 1 litro de agua, teniendo como operaciones permitidas trasvasar el contenido de un jarro a otro, operación que finaliza al llenarse el jarro de destino o vaciarse el jarro de origen.

Todo esto se tiene que lograr con el menor costo posible, siendo 1 el costo de trasvasar el contenido del jarro 1 a otro jarro.

En este caso concreto se tienen 4 jarros.

Búsqueda - Dota 0.02

Esta es una versión muy simplificada del juego Dota. En esta versión, el mapa es un tablero de 3x3, participa un solo jugador, y tiene como objetivos derrotar a un héroe enemigo y destruir su base en la menor cantidad de acciones posibles. El jugador comienza en la esquina inferior izquierda del mapa, y en cada turno puede moverse a los casilleros limítrofes (**no** en diagonal). Si se encuentra **adyacente** a un casillero de un enemigo o edificio, también tiene disponible la acción de atacar a dicho objeto, destruyéndolo como resultado. Dos objetos **no** pueden estar en la misma posición.

Al inicio del juego, el mapa y la distribución de objetos es la siguiente:

		Be
	He	
H		

H = Héroe (jugador), He = Héroe enemigo, Be = Base enemiga

Ejemplo de acciones disponibles:

		Be
H	He	

En este estado, el héroe está junto al héroe enemigo. En esta situación puede moverse en dos direcciones o atacar a He. Si ataca, en el estado resultante el héroe enemigo estaría muerto.

Hnefatafl



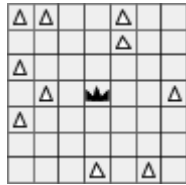
Hnefatafl es un antiguo juego vikingo similar al ajedrez, en el que un jugador tiene un pequeño ejército con un rey que debe escapar, mientras el otro jugador tiene un ejército de más fichas y su objetivo es capturar al rey antes de que escape.

En este problema vamos a resolver una versión super simplificada, en la que los soldados se encuentran fijos (no se mueven), y el rey está solo intentando escapar sin ser

atacado.

En esta versión el tablero tiene dimensiones de 7x7, el rey comienza en el casillero central, y los soldados se encuentran en las posiciones del diagrama "Posiciones iniciales". El jugador debe lograr que el rey llegue a cualquiera de los bordes del tablero en la menor cantidad posible de movimientos, pero sin ser atacado por ninguno de los soldados en

ningún momento (incluso cuando llega al borde). Los soldados atacan desde casillas adyacentes (**no** en diagonal). Nunca puede haber dos fichas en la misma posición, y los movimientos solo son a casilleros adyacentes (arriba, abajo, izquierda o derecha, **no** en diagonal, de a 1 casillero).



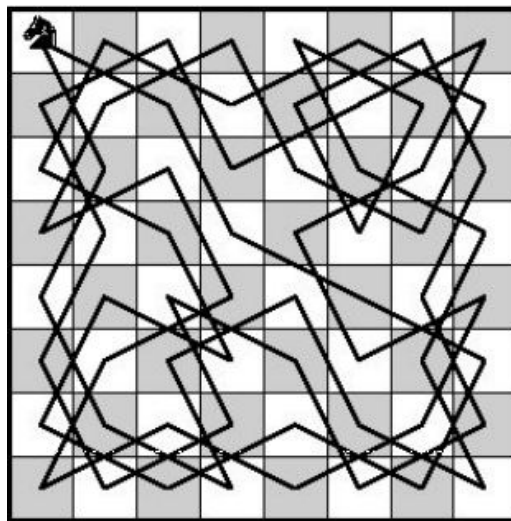
Posiciones iniciales.



Distintas formas de matar al rey.

El tour del Caballero

En el problema “Tour del Caballero” se trata de encontrar una secuencia de movimientos de una sola pieza (caballo) en un tablero de ajedrez, de modo que se visiten todas las casillas una sola vez. El “Caballero” puede comenzar en cualquier posición y hay que moverse según las reglas normales para mover un caballo en el juego de ajedrez (en forma de L).



Monos y bananas

El problema se trata de un mono en una habitación; algunas bananas cuelgan del techo y el mono desgraciadamente no puede llegar a ellas. En la sala hay una silla con la suficiente altura como para que si el mono se sube a ella pueda alcanzar las bananas. El mono puede moverse, empujar la silla y/o subirse a ella. Por supuesto que el mono quiere comer las bananas.

AYUDA: Se puede pensar que la habitación es muy estrecha (del tamaño de la silla), de 10 metros de largo, y los movimientos que realiza el mono son siempre iguales (50 cm).

Problema del Puente .

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 sólo dispone de una linterna, a la que le quedan 5 minutos de batería. Cada persona tarda en cruzar 10, 30, 60, 80 y 120 segundos, respectivamente. El puente sólo resiste un máximo de 2 personas cruzando a la vez, y cuando cruzan dos personas juntas caminan a la velocidad del más lento. 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 faltan, y así hasta que hayan cruzado todos.

Caballo y pintura .

Supongamos que tenemos un tablero rectangular dividido en cuadrados, cuyo tamaño es 4 cuadrados de largo por 3 de ancho. Inicialmente, todos los cuadrados están pintados de blanco

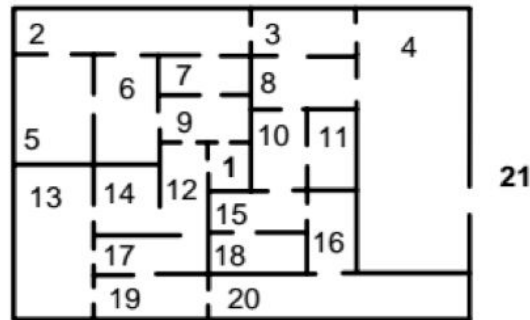
X			

El objetivo es pintar todos los cuadrados de rojo. Para ello, disponemos de un robot que es capaz de pintar cuadrados individuales de rojo, con las siguientes restricciones:

- El robot se mueve por el tablero de cuadrado en cuadrado, con movimientos simples iguales a los del caballo en el ajedrez.
- El robot nunca se puede situar sobre un cuadrado pintado de rojo.
- Una vez que ha realizado el movimiento, el robot se coloca sobre el cuadrado correspondiente (que ha de estar pintado de blanco) y lo pinta de color rojo.
- Inicialmente, el robot se encuentra sobre el cuadrado de la esquina superior izquierda, que ya está pintado de rojo.

Salir del Laberinto.

La siguiente imagen representa la estructura de un laberinto. Se debe encontrar la salida desde el casillero 1 hasta el 21.



Nave alienígena y password.

Una nave alienígena se ha estrellado en la Tierra, y un grupo de investigadores desea abrir su escotilla principal. Pero la misma posee un mecanismo bastante extraño, que consta de una serie de botones que deben ser presionados en orden para lograr una combinación de números en una pantalla.

La pantalla posee 3 casillas alineadas una al lado de la otra, cada una con un número dentro. Inicialmente todas poseen el número cero. Los botones que pueden presionarse son los siguientes:

- Botón rojo: Suma 3 al casillero inicial.
- Botón verde: Resta 2 al casillero inicial.
- Botón amarillo: Intercambia los valores de las dos primeras casillas.
- Botón celeste: Intercambia los valores de las dos últimas primeras casillas.

La secuencia de números que se debe lograr para abrir la escotilla es la siguiente: 5, 1, 8

- a. Plantee formalmente como problema de búsqueda tradicional, con **código y comentarios** explicativos.
- b. Plantee una heurística para el mismo. No hace falta que sea muy precisa, pero **debe ser admisible**.

Realice al menos 5 iteraciones de búsqueda A*. Por cada iteración especifique la frontera con el valor de f asociado a cada nodo, y el nodo elegido.

Trasladando presidentes

Durante una conferencia muy importante en las Naciones Unidas, 6 presidentes de los países más poderosos tienen que ser trasladados desde el auditorio hacia la sala de conferencias, pasando por el hall de entrada.

Pero no hay escolta suficiente para trasladarlos a todos juntos, por lo que deberán ir moviendo de a 1 o 2 presidentes por vez, y de a una habitación por movimiento, avanzando (nunca retrocediendo). Cuando los escoltas mueven a un grupo presidentes, pueden irse corriendo a cualquier habitación para mover al siguiente grupo, su tiempo de movimiento es mucho menor por lo que no nos interesa modelarlo. Podemos olvidarnos del movimiento que los escoltas solos, y pensar solo en los movimientos de los presidentes.



Los presidentes pertenecen a 3 “facciones” que no se llevan muy bien entre sí: 2 capitalistas, 2 comunistas y 2 centristas. El equipo de escolta conoce estas divisiones, y entiende que son un problema a la hora de trasladar a los mandatarios:

- Nunca pueden quedar dos presidentes de la misma facción solos en la misma sala, sin presidentes de otras facciones, porque el resto va a pensar que están conspirando.
 - Cuando dos presidentes de la misma facción están separados, solo puede ser en salas adyacentes. No puede quedar un presidente en cada extremo si son de la misma facción, porque se sentirían muy aislados de su aliado.
-
- a. Defina formalmente el problema para ser resuelto mediante búsqueda. Escriba código Python y donde considere necesario agregue comentarios que esclarezcan el funcionamiento del código.
 - b. Plantee y programe una heurística **admisible** para ser utilizada en este problema. (no importa la calidad de la misma)
 - c. Resuelva 5 iteraciones mediante búsqueda A* considerando la heurística planteada en b. Indique en cada iteración la frontera al iniciar la iteración, y el nodo elegido.

Búsqueda local.

CSP.

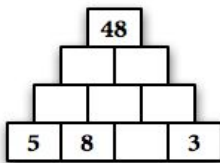
Robots defensivos secretos. (continúa...)

Luego del último ataque, se decidió incrementar el número de robots a 6, y ya no permanecerán almacenados, sino que se ubicarán en posiciones defensivas fijas y permanentes. Pero deben respetarse algunas restricciones:

- No puede haber dos robots en la misma habitación, generarían demasiadas molestias para los científicos.
- No puede haber dos robots en habitaciones adyacentes, impedirían demasiado la circulación.
- Las habitaciones restringidas siguen sin poder contener robots.
- Las dos habitaciones que poseen puertas al exterior **deben** contener un robot.

¿En qué posiciones se debieran colocar los robots?

Pirámide numérica



Dada la pirámide de números de la imagen, se desea rellenar los casilleros utilizando números del 1 al 50, de forma tal que cada casillero tenga el valor resultante de la suma de sus dos casilleros inferiores, y que no haya dos casilleros con el mismo número.

Munchkin

En el juego Munchkin, cada jugador tiene un personaje al que puede mejorar aplicando diferentes cartas. Estas cartas incluyen cosas como armas, armaduras, pociones, maldiciones, y otros modificadores que incrementan el nivel del personaje, haciéndolo más capaz de ganar el juego. Pero existen algunas restricciones respecto a qué cartas pueden utilizarse en un mismo personaje, de forma de evitar jugadores “superpoderosos” que degradan la jugabilidad.

Un jugador tiene que elegir 3 cartas para intentar armar su personaje, y las opciones disponibles son:

- | | |
|------------------------------------|-------------------------------------------|
| • Armadura de madera +1 (800 oro) | • Espada de acero +4 (1000 oro) |
| • Armadura de hierro +3 (1000 oro) | • Garrote gigante de madera +6 (1300 oro) |
| • Armadura de acero +5 (1300 oro) | • Poción de fuego +5 (1500 oro) |
| • Espada de madera +1 (500 oro) | • Poción de hielo +2 (800 oro) |
| • Espada de hierro +2 (700 oro) | • Poción de ácido +3 (1200 oro) |

Y a la vez, deben cumplirse las siguientes restricciones:

- Solo se puede tener 1 armadura
- Solo se puede tener 1 arma de mano (espada o garrote)
- Solo se dispone de 3000 de oro para gastar (es decir, el valor de las cartas sumadas no puede superar ese monto)

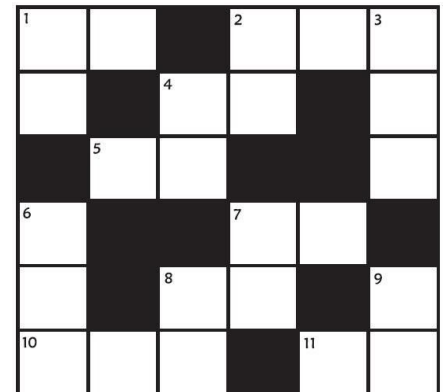
- No se pueden mezclar cartas de objetos de fuego con cartas de objetos de madera
- Se tiene que lograr un bonificador total (sumando las cartas) mayor a +8

¿Qué 3 cartas puede elegir el jugador para equipar a su personaje?

Crucigramas

Se desea implementar un sistema para generar crucigramas como el mostrado en la figura.

Esto es, dado el esquema de la derecha y un diccionario de palabras y definiciones, el algoritmo debe elegir un conjunto de palabras que podrían utilizarse para la confección del crucigrama. Para la implementación descargue algún listado de palabras de Internet.



Dota 0.02 (continúa...)

En el dota 0.02, el héroe tiene la posibilidad de comprar ítems que mejoran sus habilidades de combate. Pero algunos de esos ítems son demasiado costosos, o tienen consecuencias negativas si se combinan con otros.

Se desea elegir 3 ítems para el héroe, de la siguiente lista de posibles ítems:

- Assault cuirass: armadura que acelera los ataques (costo: 5000).
- Battlefury: hacha que hace mucho daño, daña a enemigos cercanos, y regenera vida (costo: 4000).
- Cloak: capa con resistencia a conjuros (costo: 500).
- Hyperstone: piedra que acelera muchísimo los ataques (costo: 2000).
- Quelling blade: hacha que mejora levemente el daño (costo: 200).
- Shadow blade: espada que acelera los ataques y mejora el daño (costo: 3000).
- Veil of discord: capa que regenera vida y mejora a nivel general al personaje (costo: 2000).

Pero en la elección se deben respetar las siguientes condiciones:

- Se dispone de 6000 monedas de oro para gastar en total, los ítems comprados no pueden superar esa suma.
- Hyperstone no se puede utilizar junto con Shadow blade, porque sus efectos no se suman.
- Quelling blade y Shadow blade no pueden utilizarse juntas, porque sus efectos no se suman.
- Cloak y Veil of discord no pueden utilizarse juntas, porque una es componente de la otra.
- Como mínimo se debe tener 1 ítem que regenere vida.
- Los ítems son únicos, no se puede tener dos veces el mismo ítem.

Hnefatafl (continúa...)

Se desea resolver el siguiente problema: dado el mismo tablero de 7x7, **llenar** el tablero de reyes y soldados, pero respetando las siguientes condiciones: un rey nunca puede tener

más de 1 (otro) rey en sus casillas adyacentes, y la cantidad total de soldados debe **mayor** que la cantidad de reyes, pero **menor** al doble de la cantidad total de reyes (ej: si hay 10 reyes, puede haber 15 soldados , pero no 5 soldados, 20 soldados, ni 30 soldados).

Procesos y procesadores

Se desea realizar un programa que pueda organizar un conjunto de tareas de procesamiento en un conjunto de servidores de alto rendimiento, de forma de poder ejecutar todas las tareas en paralelo, aprovechando así el poder de cómputo global. Las tareas no pueden encolarse una detrás de otra, sí o sí deben ejecutarse en paralelo y sin compartir recursos, por lo que por ejemplo no es posible ejecutar en el mismo servidor tareas que sumen más procesadores que los que el servidor posee (lo mismo sucede con todos los demás recursos).

Los servidores son los siguientes:

- Tesla: 8 procesadores, 32 GB de RAM.
- Goedel: 4 procesadores, 16GB de RAM, aceleradora gráfica.
- Bohr: 4 procesadores, 16GB de RAM.

Y las tareas a ejecutar son las siguientes:

- Limpiador de datos: requiere 2 procesadores y 10 GB de RAM.
- Convertidor de entradas: requiere 5 procesadores y 20 GB de RAM.
- Entrenador de modelos: requiere 2 procesadores, 14 GB de RAM y una aceleradora gráfica.
- Almacenador de estadísticas: requiere 1 procesador y 1GB de RAM.
- Graficador de resultados: requiere 2 procesadores y 2 GB de RAM.
- Servidor de API: requiere 2 procesadores y 8 GB de RAM.

Plantee formalmente como problema de satisfacción de restricciones, con **código y comentarios** explicativos, definiendo variables, dominios y restricciones..