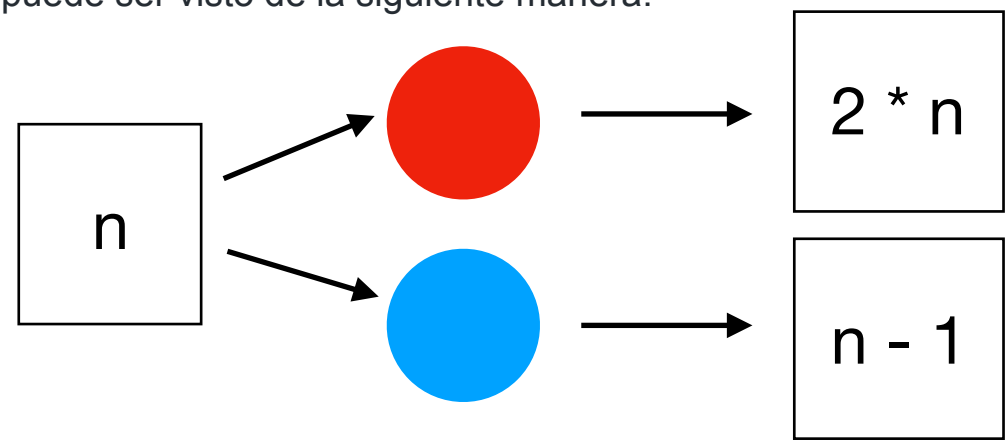
Editorial de los problemas del Club de Algoritmia CUCEI Intermedios 2019-B (Grafos D:)

Concurso: https://vjudge.net/contest/324800

Martes 1 de Octubre del 2019

A. Creo que yo soy el más fácil :D

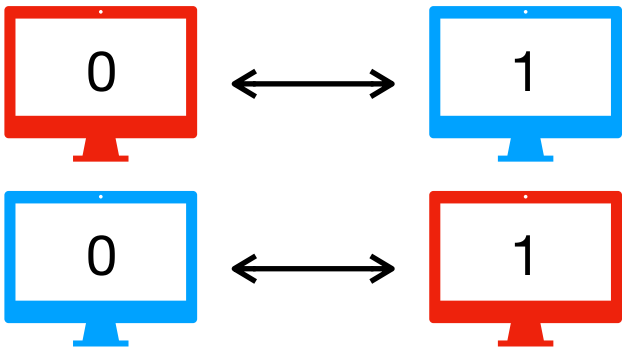
El problema nos pide saber si de un número n, se puede llegar a un número m, y de ser así, cuál es la cantidad de pasos necesarios para llegar a ese número m.
Esto puede ser visto de la siguiente manera:



Con lo anterior se puede ver claramente el grafo implícito existente, ahora sólo basta aplicar una bfs y sólo hacer los estados que no han sido realizados anteriormente, hasta lograr cumplir con la m objetivo.

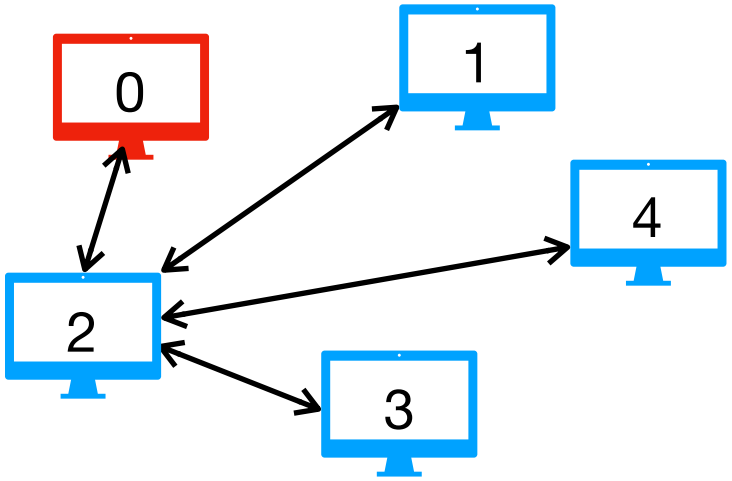
B. ¿O soy yo el más fácil?

El problema nos pide decir la distancia óptima para colocar el router, de tal manera que esta sea la máxima de las distancias mínimas a los demás nodos, es decir, si se probara cada nodo como una ubicación posible del router, la máxima distancia de las distancias mínimas sería la respuesta para este acomodo del router en ese nodo; se buscar minimizar lo anterior.
Para el primer caso de ejemplo, usando la computadora 0 (rojo) como el router, la distancia mínima máxima es 1 (0->1) y viceversa tomando al 1 (rojo) como nuevo router, con distancia mínima máxima 1 (1->0).

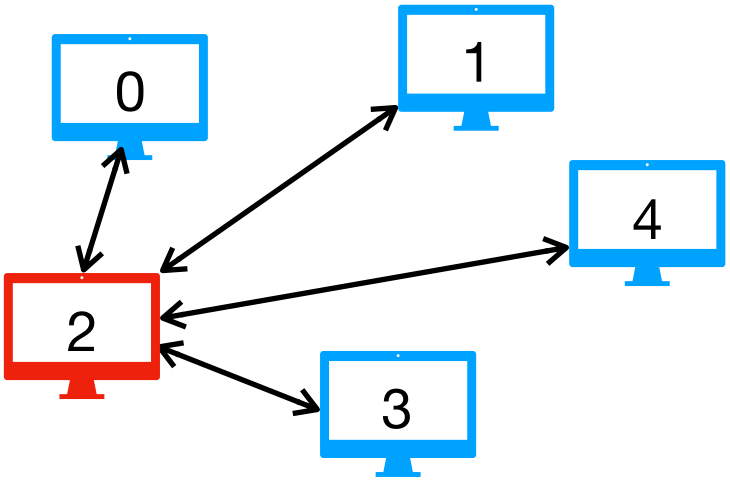


Como nota, el ejemplo sólo nos sirve para ilustrarnos qué debemos de hacer, pero no nos ayuda a la solución.

De este ejemplo, tomando todas las distancias mínimas y calculando la máxima de estas nos queda lo siguiente (probando sólo dos)

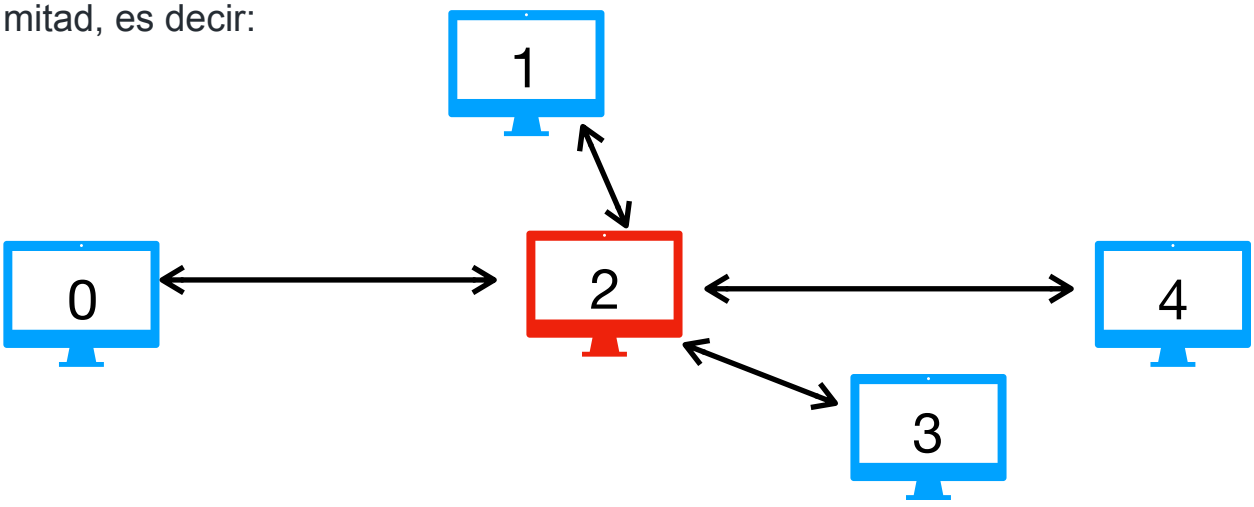


0->2->1 = 2
0->2 = 1
0->2->4 = 2
0->2->3 = 2
La máxima sería 2 si escogemos a 0 como router



2->0 = 1
2->1 = 1
2->4 = 1
2->3 = 1
La máxima sería 1 si escogemos a 2 como router

Como podemos notar si ecogemos cualquier otro nodo {1, 3, 4}, sería como si hubiéramos tomado al nodo 0, es decir, no óptimo, por análisis, debemos de agarra un nodo tal que se encuentre al centro del grafo (algo avanzado de calcular), pero al sólo pedirnos la distancia, podemos calcular a qué distancia puede quedar un nodo si lo alargamos y lo colocamos a la mitad, es decir:



Como podemos notar, el nodo 2 con la distancia máxima de todas las mínimas, se encuentra en el centro de el camino más largo entre dos nodos (diámetro), por lo que podemos calcular el radio del mismo (diámetro / 2) y está sería nuestra respuesta.

C . ¡No!, Yo soy el más fácil

Lo primero que podemos notar es que se nos da una matriz de adyacencia y se requiere ir de un nodo al mismo nodo con la menor distancia (un ciclo), al ser una cantidad muy baja de nodos podemos probar para cada nodo u intentar llegar a u, y de ser posible hemos encontrado una respuesta, cabe destacar que esta solución tiene una complejidad cúbica N por probar cada nodo y N^2 por revisar la adyacencia de cada nodo, por lo que nos dará un TLE, sin embargo podemos optimizar esa complejidad a N por probar cada nodo y M creando en base a nuestra matriz de adyacencia una lista de adyacencia por lo que nuestra complejidad podrá entrar en el tiempo especificado.
Como implementar cada bfs, únicamente es necesario tener la distancia y el nodo actual y si estamos nuevamente en el nodo desde el que empezamos y tenemos una distancia mayor a 0, quiere decir que realmente nos movimos.

D . ¿Algún día seré el más fácil?

El problema nos pide dar una secuencia de apertura de cajas para obtener llaves del resto de tal manera que sea la más corta posible (alcancemos el tesoro lo más rápido posible) y sea la menor lexicográficamente (esto puesto que pueden existir varias soluciones de apertura de cajas que cumplan con el objetivo), esto suena algo aterrador, pero si notamos bien la bfs nos permite irnos en orden, por lo que si agarramos un elemento a < b, donde ambos a y b son parte de nuestra solución, b jamás volverá a ser procesado (a menos que sea mejor), puesto que a hizo exactamente lo mismo y tiene un valor menor que b, es decir, podemos ordenar nuestra lista de adyacencia y con esto siempre garantizaremos tener la solución más pequeña lexicográficamente.
Dado lo anterior, sólo nos queda encontrar una manera de reconstruir el camino realizado, para esto para cada posición posible guardaremos el estado del que vino anteriormente (el nodo posterior a llegar) y sólo tendremos que recorrerlo desde e nodo objetivo hasta el nodo inicio y tendremos nuestra solución.

E - Yo no soy el más fácil D:

Dado que siempre tenemos incrementos diferentes, no es posible resolverlo con una bfs, pero sí con un dijkstra en base a los dólares que tiene que pagar por cada movimiento realizado y hemos terminado con el problema. Hay que notar que nos podemos mover como una caballo de ajedrez, y que el tablero siempre es de 8 x 8, por lo que siempre que se haga un dijsktra apropiadamente entrará en tiempo.