

Soluciones entrenos OIE

Cave

Te has desviado accidentalmente del camino que hay entre la facultad y el UQ Centre, y has dado con la entrada a una cueva secreta que alguien construyó bajo el campus universitario. La entrada está bloqueada por un sistema de seguridad formado por N puertas consecutivas, cada una a continuación de la anterior, y N interruptores, cada uno de ellos conectado a una puerta diferente.

Las puertas están numeradas como $0, 1, \dots, (N - 1)$ en orden, donde la puerta 0 es la que tienes más cerca. Los interruptores también están numerados como $0, 1, \dots, (N - 1)$, pero no sabes a qué puerta está conectado cada interruptor.

Todos los interruptores se encuentran en la entrada de la cueva. Cada interruptor puede estar en una de las dos posiciones siguientes: *subido* o *bajado*. Por cada uno de los interruptores hay una única posición correcta. Si un interruptor está en su posición correcta entonces la puerta a la que está conectado estará abierta, mientras que si el interruptor está en su posición incorrecta entonces la puerta a la que está conectado estará cerrada. Dos interruptores distintos pueden tener posiciones correctas diferentes, e inicialmente no sabes cuáles son esas posiciones correctas.

Tienes curiosidad por entender el sistema de seguridad de la cueva. Para ello puedes cambiar la posición de los interruptores, dejándolos en cualquier combinación posible, y a continuación entrar en la cueva para ver cual es la primera puerta que se encuentra cerrada. Las puertas no son transparentes: una vez que encuentres la primera puerta cerrada no puedes ver ninguna de las puertas que hay detrás de esta. Sólo tienes tiempo para probar 70000 combinaciones de interruptores. Tu misión es determinar la posición correcta de cada interruptor, y también a qué puerta está conectado.

Restricciones

$$1 \leq N \leq 5000$$

Subtareas

- 12 Puntos: El interruptor i está conectado a la puerta i .
- 13 Puntos: Las posiciones correctas de los interruptores son todas 0.
- 21 Puntos: $N \leq 100$
- 30 Puntos: $N \leq 2000$
- 24 Puntos: (Sin restricciones adicionales)

Solución

Primero intentemos deducir cuál es el interruptor correspondiente a la primera puerta y cuál es su posición correcta. Para encontrar la posición correcta, podemos poner todos los interruptores en la misma posición y hacer una consulta para ver si podemos pasar de a primera puerta: si podemos, entonces esa será la posición correcta y si no podemos será la posición contraria.

Para encontrar cuál es el interruptor conectado a esa puerta podemos hacer una búsqueda binaria sobre los interruptores: empezamos poniendo la mitad en la posición correcta y la otra mitad en la posición incorrecta, hacemos una consulta y así descartamos la mitad, luego la mitad de los que nos quedan los ponemos en la posición correcta y los otros en posición incorrecta, etcétera. Al final encontraremos el interruptor correspondiente a la primera puerta en $\approx \log_2 N$ consultas

La razón por la que esto funciona para la primera puerta y no para las demás es porque cuando haces la consulta, si hay una puerta anterior a la puerta que estás estudiando que está cerrada entonces con la consulta ya no puedes determinar el estado de la puerta que quieres. Pero si te aseguras de que todas las puertas anteriores estén abiertas, entonces puedes realizar la misma estrategia: lo único diferente que hay que hacer es que, en lugar de ir cambiando las posiciones de todos los interruptores, los de las puertas anteriores los tienes que dejar siempre en la posición correcta. Así, podemos ir determinando uno por uno los interruptores de cada puerta, haciendo un total de $N \log_2 N$ consultas, que es un poco menos de 70000 para $N = 5000$.

Código

C++

```
1  #include "cave.h"
2
3
4  //position: guarda la posicion correcta de cada interruptor
5  int position[5000];
6  //S: array que usamos para las preguntas
7  int S[5000];
8  //door: guarda la puerta asociada a cada interruptor
9  int door[5000];
10 //Tenemos n como variable global por comodidad
11 int n;
12
13 //Usamos esta funcion para preguntar y asi siempre que preguntemos
14 //ponemos los interruptores ya conocidos en la posicion correcta
15 int ask() {
16     for(int i=0; i < n; ++i) {
17         if(position[i] != -1) S[i] = position[i];
18     }
19     return tryCombination(S);
20 }
21
22 void exploreCave(int N) {
23     n = N;
24
25     //Inicialmente no conocemos ninguna posicion
26     for(int i=0; i < n; ++i) position[i] = -1;
```

```

27
28 for(int i=0; i < n; ++i) {
29     //Determinamos el interruptor de la i-esima puerta
30     for(int j=0; j < n; ++j) S[j] = 0;
31
32     //Hacemos la primera pregunta para determinar la posicion
33     //cur_pos guarda la posicion correcta
34     int cur_pos;
35     int res = ask();
36     //recordemos que si da -1 es que todas las puertas estan abiertas
37     if(res > i or res == -1) {
38         cur_pos = 0;
39     }
40     else {
41         cur_pos = 1;
42     }
43
44     //busqueda binaria: buscamos en el intervalo [low, high)
45     int low = 0;
46     int high = n;
47     while(high > low+1) {
48         int mid = (low+high)/2;
49         //ponemos [low, mid) en posicion incorrecta y
50         //[mid, high) en correcta
51         for(int j=low; j < high; ++j) {
52             if(j < mid) S[j] = 1-cur_pos;
53             else S[j] = cur_pos;
54         }
55         res = ask();
56         if(res > i or res == -1) {
57             low = mid;
58         }
59         else {
60             high = mid;
61         }
62     }
63
64     position[low] = cur_pos;
65     door[low] = i;
66 }
67 answer(position, door);
68 }

```