



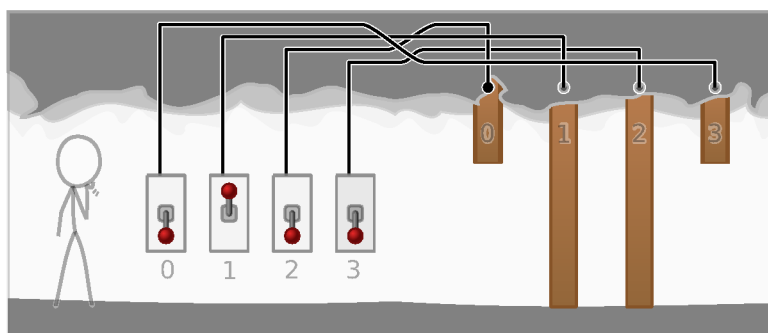
International Olympiad in Informatics 2013

6-13 July 2013
Brisbane, Australia
Day 2 tasks

jaskyňa

Slovak — 1.0

Počas dlhej prechádzky z internátu do centra Queensland-skej univerzity ste sa stratili a náhodou ste objavili vchod do tajného systému jaskýň pod univerzitou. Vstup je blokovaný systémom pozostávajúcim z N za sebou nasledujúcich dverí a N vypínačov, pričom každý vypínač je spojený s inými dverami.



Dvere sú očíslované v poradí $0, 1, \dots, (N - 1)$, pričom dvere s číslom 0 sú najbližšie ku vám. Vypínače sú tiež očíslované $0, 1, \dots, (N - 1)$, pričom ale neviete, ktorý vypínač ku ktorým dverám patrí.

Vypínače sú umiestnené pri vstupe do jaskyne. Každý vypínač môže byť v jednej z dvoch polôh - "zapnutý" alebo "vypnutý". Ak je vypínač v správnej polohe, dvere pripojené k nemu sa otvoria. Ak je v nesprávnej polohe, tak dvere zostanú zatvorené. Správna poloha je pre rôzne vypínače rôzna a vy nepoznáte, ktoré polohy sú správne.

Rozhodli ste sa, že detailne zistíte, ako tento systém funguje. Môžete si zvoliť ľubovoľnú kombináciu polôh vypínačov a pozrieť sa do jaskyne, ktoré dvere vidíte ako prvé. Dvere sú nepriehľadné, takže nemôžete vidieť žiadne ďalšie dvere za nimi.

Môžete vyskúšať najviac 70.000 kombinácií polôh vypínačov. Vašou úlohou je zistiť správnu polohu pre každý vypínač, a takisto ku ktorým dverám je pripojený.

Implementácia

Odovzdajte súbor, v ktorom implementujete procedúru `exploreCave()`. Tá môže zavolať najviac 70.000 krát funkciu testovača `tryCombination()` a musí skončiť zavolaním procedúry testovača `answer()`. Tieto funkcie a procedúry sú popísané nižšie.

funkcia testovača: `tryCombination()`

C/C++ `int tryCombination(int S[]);`

Pascal `function tryCombination(var S: array of LongInt) : LongInt;`

Popis

Táto funkcia je súčasťou testovača. Dovolí vám vyskúšať ľubovoľnú kombináciu a vráti číslo prvých zatvorených dverí. V prípade, že sú všetky dvere otvorené, funkcia vráti hodnotu `-1`. Táto funkcia beží v čase $O(N)$.

Túto funkciu môžete zavolať najviac 70.000 krát.

Parametre

- `S`: Pole dĺžky `N`, reprezentujúce polohu každého vypínača. Prvok `S[i]` udáva polohu vypínača `i`. Hodnota `0` znamená polohu zapnutý a hodnota `1` znamená polohu vypnutý.
- *Návratová hodnota*: Číslo prvých zatvorených dverí, resp. `-1` v prípade, že sú všetky dvere otvorené.

procedúra testovača: `answer()`

C/C++ `void answer(int S[], int D[]);`

Pascal `procedure answer(var S, D: array of LongInt);`

Popis

Zavolajte túto procedúru, ak už budete poznať správnu kombináciu na otvorenie všetkých dverí a zároveň priradenie všetkých vypínačov k dverám.

Parametre

- `S`: pole dĺžky `N` reprezentujúce správne polohy každého vypínača. Formát je rovnaký ako vo funkcii `tryCombination()` popísanej vyššie.
- `D`: pole dĺžky `N` reprezentujúce čísla dverí prislúchajúce jednotlivým vypínačom. Presnejšie, prvok `D[i]` má obsahovať číslo dverí pripojených k vypínaču `i`.
- *Návratová hodnota*: Táto procedúra ukončí program.

vaša procedúra: `exploreCave()`

C/C++ `void exploreCave(int N);`

Pascal `procedure exploreCave(N: longint);`

Popis

Musíte implemetovať túto procedúru.

Táto procedúra má použiť funkciu testovača `tryCombination()` na určenie správnej polohy každého vypínača, priradenie vypínačov k dverám a musí raz zavolať `answer()`, keď už tieto informácie zistíte.

Parametre

- `N`: počet dverí a vypínačov v jaskyni.

Ukázkový beh

Majme dvere a vypínače priradené ako na obrázku vyššie:

volanie funkcie	návratová hodnota	vysvetlenie
<code>tryCombination([1, 0, 1, 1])</code>	1	Popisuje príklad z obrázku. Vypínače 0, 2 a 3 sú vypnuté a vypínač 1 je zapnutý. Táto funkcia vráti 1, tzn. že prvé zatvorené dvere sú dvere číslo 1.
<code>tryCombination([0, 1, 1, 0])</code>	3	Dvere 0, 1 a 2 sú všetky otvorené, dvere 3 sú zatvorené.
<code>tryCombination([1, 1, 1, 0])</code>	-1	Zapnutie vypínača 0 otvorí všetky dvere, čo znamená návratovú hodnotu -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	(ukončenie programu)	Zistili sme správnu kombináciu [1, 1, 1, 0] a vypínače 0, 1, 2 a 3 sú postupne pripojené k dverám 3, 1, 0 a 2.

Obmedzenia

- Časový limit: 2 sekundy
- Pamäťový limit: 32 MiB
- $1 \leq N \leq 5,000$

Podúlohy

Podúloha	Body	Ďalšie obmedzenia
1	12	Pre každé i , vypínač i je prepojený s dverami číslo i . Vašou úlohou je teda určiť správnu kombináciu.
2	13	Správna kombinácia je <code>[0, 0, 0, ..., 0]</code> . Vašou úlohou je teda určiť správne priradenie vypínačov k dverám.
3	21	$N \leq 100$
4	30	$N \leq 2,000$
5	24	(žiadne)

Experimentácia

Ukážkový testovač na vašom počítači bude čítať vstup zo súboru `cave.in`, ktorý musí mať nasledovný formát:

- riadok 1: `N`
- riadok 2: `S[0] S[1] ... S[N - 1]`
- riadok 3: `D[0] D[1] ... D[N - 1]`

`N` určuje počet dverí a vypínačov, `S[i]` správne polohy vypínačov `i` a `D[i]` čísla dverí pripojených k vzpínaču `i`.

Napríklad príklad uvedený vyššie, by bol zapísaný nasledovne:

```
4
1 1 1 0
3 1 0 2
```

Poznámky k programovacím jazykom

C/C++ Musíte použiť `#include "cave.h"`.

Pascal Musíte definovať `unit Cave`, a takisto musíte importovať knižnicu `uses GraderHelpLib`. Všetky polia sú očíslované od `0` (nie od `1`).

Vid' predlohy riešení na vašom počítači.