



International Olympiad in Informatics 2013

6-13 July 2013

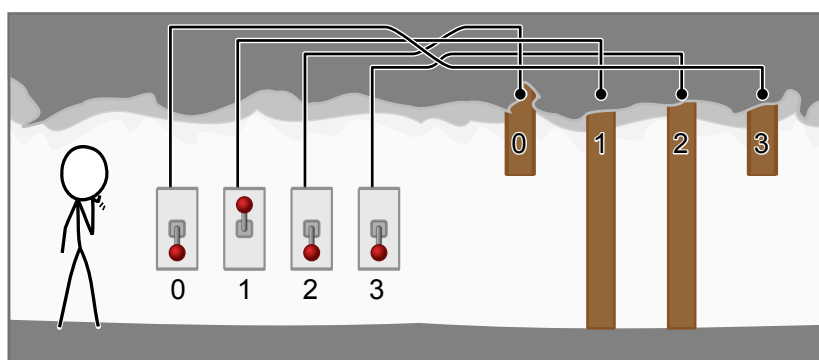
Brisbane, Australia

Day 2 tasks

cave

Bosanski — 1.0

Tokom duge šetnje kroz UQ koledž, naišli ste na ulaz u sakriveni sistem podzemnih pećina ispod univerzitetskog kompleksa. Ulaz u pećine blokiran je sigurnosnim sistemom koji se sastoji od N uzastopnih vrata. Vrata su poredana jedna iza drugih. Postoji N prekidača - po jedan prekidač za svaka od vrata.



Vrata su označena brojevima $0, 1, \dots, (N - 1)$, pri čemu su vrata 0 Vama najbliža. Prekidači su također označeni brojevima sa $0, 1, \dots, (N - 1)$, ali ne znate koji prekidač otvara koja vrata.

Svi prekidači su postavljeni na ulaz u pećinu. Svaki od njih može biti u jednom od dva položaja *gore* ili *dole*. Samo jedan od ovih položaja je ispravan. Ako je prekidač u ispravnom položaju, tada su njemu odgovarajuća vrata otvorena. U suprotnom, ako je prekidač u pogrešnom položaju, tada su vrata koja prekidač otvara, zatvorena. Ispravan položaj prekidača može se razlikovati za različite prekidače, i Vama nije unaprijed poznato koji od dva položaja je ispravan za koji prekidač.

Vaš cilj je da razbijete ovaj sigurnosni sistem. S tim ciljem, možete da postavite bilo koju kombinaciju položaja prekidača, a onda uđete u pećinu da biste ustanovili koja su prva zatvorena vrata. Vrata nisu providna, odnosno kada naidete na prva zatvorena vrata, ne možete vidjeti ništa iza njih.

Imate dovoljno vremena da probate **70.000** kombinacija položaja prekidača, ali ne više od toga. Vaš zadatak je da odredite ispravno stanje za svaki prekidač, te da odredite koji su prekidači spojeni sa kojim vratima.

Implementacija

Treba da submitujete file sa implementiranom procedurom `exploreCave()`. Ova procedura smije pozvati najviše 70.000 puta funkciju `tryCombination()` koja je implementirana u grader-u, i treba završiti izvršavanje pozivom procedure `answer()` koja je također implementirana u grader-u. Ove funkcije i procedure su opisane ispod.

Funkcija implementirana u grader-u: `tryCombination()`

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

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

Opis

Ova funkcija je implementirana u grader-u. Omogućava Vam da izaberete kombinaciju položaja prekidača, a zatim ulaskom u pećinu otkrijete koja su prva zatvorena vrata. Ako su za izabranu kombinaciju prekidača sva vrata otvorena onda funkcija vraća `-1`. Vrijeme izvršavanja ove funkcije je $O(N)$, odnosno u najgorem slučaju je proporcionalno sa `N`.

Ovu funkciju smijete pozvati najviše `70.000` puta.

Parametri

- `S`: Niz dužine `N` kojim birate položaj svakog prekidača. Element `S[i]` odgovara prekidaču `i`. Vrijednost `0` označava da je prekidač u poziciji *gore*, dok vrijednost `1` označava da je prekidač u poziciji *dole*.
- *Vraća*: Oznaku prvih vrata koja su zatvorena, ili `-1` ako su sva vrata otvorena.

Procedura implementirana u grader-u: `answer()`

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

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

Opis

Ovu proceduru treba da pozovete kada otkrijete kombinaciju položaja prekidača koja otvara sva vrata i kada otkrijete koji prekidač otvara koja vrata.

Parametri

- `S`: Niz dužine `N`, kojim birate ispravne položaje za svaki prekidač. Format ovog niza je isti kao u funkciji `tryCombination()` koja je opisana ranije.
- `D`: Niz dužine `N`, kojim označavate vezu svakog prekidača sa odgovarajućim vratima. Preciznije, elemenat `D[i]` treba da sadrži oznaku vrata sa kojima je povezan prekidač `i`.
- *Vraća*: Ova procedura ne vraća vrijednost, ali uzrokuje završetak izvršavanja programa.

Vaša procedura: `exploreCave()`

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

Pascal `procedure exploreCave(N: longint);`

Opis

Vaše rješenje mora da implementira ovu proceduru.

Ova funkcija treba da koristi funkciju `tryCombination()` implementiranu u grader-u da bi odredila ispravan položaj svakog od prekidača i veze prekidača sa vratima. Nakon što odredite ove informacije, obavezno pozovite proceduru `answer()`.

Parametri

- `N`: Broj prekidača i vrata u pećini.

Primjer izvršavanja

Pretpostavite da su vrata i prekidači organizovani kao na slici datoj ranije.

Poziv funkcije	Vraća	Objašnjenje
<code>tryCombination([1, 0, 1, 1])</code>	1	Ova situacija odgovara slici na početku. Prekidači 0, 2 i 3 su u položaju <i>dole</i> , dok je prekidač 1 u položaju <i>gore</i> . Funkcija vraća 1, što znači da su vrata 1 prva vrata (s lijeva) koja su zatvorena.
<code>tryCombination([0, 1, 1, 0])</code>	3	Vrata 0, 1 i 2 su otvorena, dok su vrata 3 zatvorena.
<code>tryCombination([1, 1, 1, 0])</code>	-1	Ako se položaja prekidača 0 promijeni na <i>dole</i> , sva vrata će biti otvorena, pa funkcija vraća -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	<i>(Izvršavanje programa se završava)</i>	Pogađamo da je ispravna kombinacija <code>[1, 1, 1, 0]</code> , i da su prekidači 0, 1, 2 i 3 povezani sa vratima 3, 1, 0 i 2 respektivno.

Ograničenja

- Vremensko ograničenje: 2 sekunde
- Memorijsko ograničenje: 32 MiB
- $1 \leq N \leq 5,000$

Podzadaci

Podzadatak	Bodovi	Dodatna ograničenja na ulazne podatke
1	12	Za svako i , prekidač i je povezan sa vratima i . Vaš zadatak je samo da odredite ispravne položaje prekidača.
2	13	Ispravna kombinacija položaja je uvijek $[0, 0, 0, \dots, 0]$. Vaš zadatak je samo da odredite koji prekidač je povezan na koja vrata.
3	21	$N \leq 100$
4	30	$N \leq 2,000$
5	24	<i>(Nema dodatnih ograničenja)</i>

Lokalno testiranje

Grader na vašem računaru čita ulazne podatke iz file-a `cave.in`, koji mora biti u ovom formatu:

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

Ovdje je `N` broj vrata i prekidača, `S[i]` je ispravan položaj za prekidač `i`, dok je `D[i]` oznaka vrata na koja je povezan prekidač `i`.

Ilustracije radi, primjer iznad bi trebao biti zapisan u sljedećem formatu:

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

Napomene vezane za programske jezike

C/C++ Morate uraditi `#include "cave.h"`.

Pascal Morate definisati `unit Cave`, te morate import-ovati rutine grader-a sa `uses GraderHelpLib`. Svi nizovi su indeksirani počevši od `0` (a ne od `1`).

Pogledajte ponuđena jednostavna rješenja na Vašem računare za primjere.