



International Olympiad in Informatics 2013

6-13 July 2013

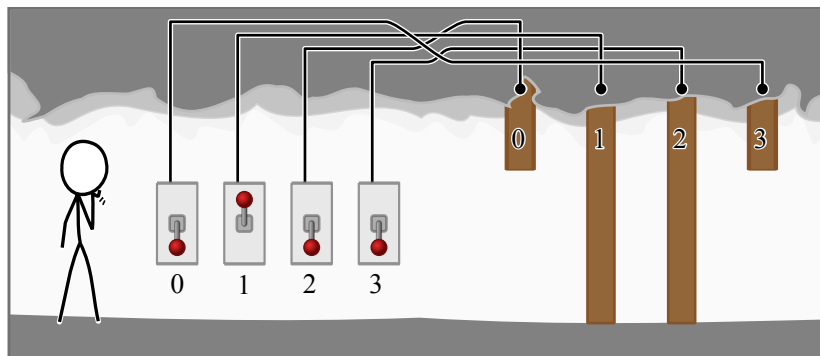
Brisbane, Australia

Day 2 tasks

cave

magyar — 1.0

Az UQ Center épületébe egymás utáni N ajtón keresztül lehet bejutni. Van N kapcsoló, mindegyik ajtóhoz 1 kapcsoló.



Az ajtók és a kapcsolók $0, 1, \dots, (N - 1)$ sorszámúak. Nem tudjuk, hogy melyik kapcsoló melyik ajtót nyitja.

Az N kapcsoló a bejáratnál van. Mindegyik kétállású: *fent* és *lent*, az egyik nyitja az ajtót, a másik zárja, de nem tudjuk, hogy melyik. A nyitó állás az egyes ajtókra különböző lehet.

Minden kapcsolót beállíthatsz valamelyik állásra és elmehetsz az első zárt ajtóig (a nyitott ajtókon keresztül). Az ajtók nem átlátszóak, az első zárt ajtó mögöttiek állapotát nem látod.

Legfeljebb 70.000 próbálkozással kell kitalálnod, hogy melyik kapcsoló melyik ajtót nyitja és milyen állással.

Megvalósítás

Az `exploreCave()` eljárást kell beküldened. Ez legfeljebb 70,000-szer hivatja a `tryCombination()` eljárást, a legvégén az `answer()` eljárással kell befejezni.

Értékelő függvény: `tryCombination()`

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

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

Leírás

Az értékelőben van ez a függvény. Amikor meghívod kapcsolók egy kombinációjával, eredményül megadja az első zárt ajtó sorszámát. Ha minden ajtó nyitott, akkor a `-1` értéket adja. A függvény futási ideje $O(N)$ vagyis N -nel arányos idejű.

Legfeljebb `70,000`-szer lehet hívni.

Paraméterek

- `S`: N elemű tömb, a kapcsolók állását tartalmazza, az `S[i]` az i . kapcsolóét. A `0` jelenti a *fent* állást, az `1` a *lent* állást.
- *Visszatérési érték*: A legelső zárt ajtó sorszáma, vagy `-1`, ha az összes ajtó nyitva van.

Értékelő eljárás `answer ()`

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

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

Leírás

Ezt az eljárást kell hívnod a megoldás közlésére.

Paraméterek

- `S`: `N` elemű tömb, a kapcsolók megfelelő állását adja. Paraméterezése megegyezik a `tryCombination ()` eljárásával.
- `D`: `N` elemű tömb, amely megadja, hogy a kapcsolók melyik ajtóhoz tartoznak. `D[i]` értéke annak az ajtónak a sorszáma, amelyhez az `i`. kapcsoló tartozik.
- *Visszatérési érték*: A program végrehajtásának befejezését eredményezi, nem tér vissza.

Az eljárásod: `exploreCave ()`

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

Pascal `procedure exploreCave(N: longint);`

Leírás

Ezt az eljárást kell beküldened.

Függvényed használja a `tryCombination ()` eljárást a megoldás kitalálásához és végül az `answer ()` eljárás meghívásával közli az eredményt.

Paraméterek

- `N`: az ajtók és a kapcsolók száma.

Példa

Az ajtók és a kapcsolók az ábra szerintiek:

Függvényhívás	Visszatérési érték	Magyarázat
<code>tryCombination([1, 0, 1, 1])</code>	1	Ez megfelel a képnek. A 0., 2. és 3. kapcsolók lent állásúak, az 1. pedig fent állású. A visszatérési érték 1, mert a legelső zárt ajtó az 1-es sorszámú lesz.
<code>tryCombination([0, 1, 1, 0])</code>	3	Ennek hatására a 0., 1. és 2. ajtók lesznek nyitva, a 3. ajtó pedig zárva.
<code>tryCombination([1, 1, 1, 0])</code>	-1	Ha az előzőhöz képest a 0. kapcsolót lent állásba állítjuk, akkor minden ajtó nyitva lesz, a visszatérési érték: -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	<i>(A program befejeződik.)</i>	A helyes kombináció: <code>[1, 1, 1, 0]</code> , és a 0., 1., 2. és 3. kapcsolók rendre a 3., 1., 0. és 2. ajtóhoz tartoznak.

Korlátok

- Időlimit: 2 másodperc
- Memória limit: 32 MiB
- $1 \leq N \leq 5,000$

Részfeladatok

Részfeladat	Pontszám	További korlátok
1	12	Az i . kapcsoló az i . ajtóhoz tartozik. Csak a kapcsolók állásait kell kitalálni.
2	13	A helyes kapcsolóállás mindig a <code>[0, 0, 0, ..., 0]</code> . Csak azt kell kitalálni, hogy melyik kapcsoló melyik ajtóhoz tartozik.
3	21	$N \leq 100$
4	30	$N \leq 2,000$
5	24	<i>(Nincs)</i>

Gyakorlás

A mintaértékelő a `cave.in` fájlból olvassa a bemenetet, amelynek formája:

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

Itt `N` az ajtók és kapcsolók száma, `S[i]` az `i`. kapcsoló helyes állása, `D[i]` az `i`. kapcsolóhoz tartozó ajtó sorszáma.

A példának az alábbi fájl felel meg:

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

Nyelvi előírások

C/C++ Iportálnod kell: `#include "cave.h"`.

Pascal Definiálni kell a `unit Cave`-t, és importálni a `uses GraderHelpLib`-t.
A tömbök indexelése `0`-tól kezdődik (nem `1`-től).

Lásd a mintaprogramokat a gépeden!