



International Olympiad in Informatics 2013

6-13 July 2013

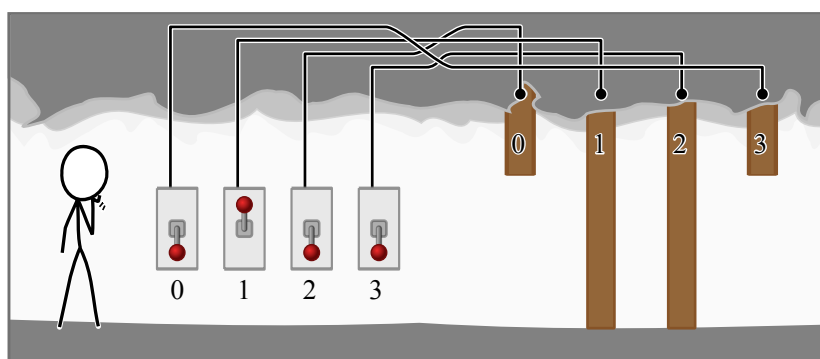
Brisbane, Australia

Day 2 tasks

cave

Norwegian — 1.0

Da du gikk deg bort på den lange gåturen fra colleget til UQ Centre snublet du over inngangen til et hemmelig hulesystem som går dypt under universitetet. Inngangen er blokkert av et sikkerhetssystem som består av N dører bak hverandre, og N brytere som er koblet til hver sin dør.



Dørene er nummerert $0, 1, \dots, (N - 1)$ i den rekkefølgen de står, hvor dør 0 er nærmest deg. Bryterne er også nummerert $0, 1, \dots, (N - 1)$, men du vet ikke hvilken bryter som er koblet til hvilken dør.

Alle bryterne befinner seg ved inngangen til hulen. Hver bryter kan enten være i *opp*- eller *ned*-posisjon. Bare en av disse posisjonene er riktig for hver bryter. Hvis en bryter er i riktig stilling vil døren den er koblet til være åpen, og hvis bryteren er i gal stilling vil døren den er koblet til være stengt. Den riktige stillingen kan variere fra bryter til bryter, og du vet ikke hvilke stillinger som er riktige.

Du vil gjerne forstå sikkerhetssystemet. For å gjøre dette kan du sette bryterne til en hvilken som helst kombinasjon, og så gå inn i hulen for å se hva som er den første stengte døren. Dørene er ikke gjennomsiktige: med en gang du støter på den første stengte døren, kan du ikke se noen av dørene som står lengre inn.

Du har tid til å prøve **70 000** bryterkombinasjoner, men ikke mer. Oppgaven din er å avgjøre den korrekte stillingen for hver bryter, og hvilken dør hver bryter er koblet til.

Implementasjon

Du må sende inn en fil som implementerer funksjonen `exploreCave()`. Denne kan kalle grader-funksjonen `tryCombination()` opptil 70 000 ganger, og må avslutte med å kalle grader-funksjonen `answer()`.

Grader-funksjon: `tryCombination()`

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

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

Beskrivelse

Graderen har implementert denne funksjonen. Den lar deg prøve en kombinasjon av brytere, og så gå inn i hulen for å finne den første lukkede døren. Hvis alle dørene er åpne, vil funksjonen returnere `-1`. Denne funksjonen kjører på $O(N)$ tid; det vil si at kjøretiden i verste fall er proporsjonal med N .

Denne funksjonen kan kalles maksimalt `70 000` ganger.

Parametre

- `S`: Et array av lengde N , som indikerer stillingen til hver bryter. Elementet `S[i]` tilsvarer bryter `i`. Verdien `0` indikerer at bryteren er oppe, og verdien `1` indikerer at bryteren er nede.
- *Returnerer*: Nummeret til den første døren som er stengt, eller `-1` hvis alle dørene er åpne.

Grader-funksjon: `answer()`

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

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

Beskrivelse

Kall denne funksjonen når du har identifisert bryterkombinasjonen som åpner alle dørene, og hvilken dør hver bryter er koblet til.

Parametre

- `S`: Et array med lengde `N`, som indikerer den korrekte stillingen for hver bryter. Formatet tilsvarer det i `tryCombination()` som er beskrevet ovenfor.
- `D`: Et array med lengde `N`, som indikerer hvilken dør hver bryter er koblet til. Mer spesifikt skal element `D[i]` inneholde dørnnummeret som bryter `i` er koblet til.
- *Returnerer*: Denne funksjonen returnerer ikke; den vil få programmet til å avslutte.

Din funksjon: `exploreCave()`

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

Pascal `procedure exploreCave(N: longint);`

Beskrivelse

Besvarelsen din må implementere denne funksjonen.

Denne funksjonen skal bruke grader-funksjonen `tryCombination()` for å avgjøre den riktige stillingen for hver bryter og hvilken dør hver bryter er koblet til, og må kalle `answer()` med en gang den har avgjort denne informasjonen.

Parametre

- `N`: Antall brytere og dører i hulen.

Sample Session

Suppose the doors and switches are arranged as in the picture above:

Function Call	Returns	Explanation
<code>tryCombination([1, 0, 1, 1])</code>	1	This corresponds to the picture. Switches 0, 2 and 3 are down, while switch 1 is up. The function returns 1, indicating that door 1 is the first door from the left that is closed.
<code>tryCombination([0, 1, 1, 0])</code>	3	Doors 0, 1 and 2 are all opened, while door 3 is closed.
<code>tryCombination([1, 1, 1, 0])</code>	-1	Moving switch 0 down causes all doors to be opened, indicated by the return value of -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	<i>(Program exits)</i>	We guess that the correct combination is [1, 1, 1, 0], and that switches 0, 1, 2 and 3 connect to doors 3, 1, 0 and 2 respectively.

Constraints

- Time limit: 2 seconds
- Memory limit: 32 MiB
- $1 \leq N \leq 5,000$

Subtasks

Subtask	Points	Additional Input Constraints
1	12	For each i , switch i is connected to door i . Your task is simply to determine the correct combination.
2	13	The correct combination will always be <code>[0, 0, 0, ..., 0]</code> . Your task is simply to determine which switch connects to which door.
3	21	$N \leq 100$
4	30	$N \leq 2,000$
5	24	<i>(None)</i>

Experimentation

The sample grader on your computer will read input from the file `cave.in`, which must be in the following format:

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

Here `N` is the number of doors and switches, `S[i]` is the correct position for switch `i`, and `D[i]` is the door that switch `i` is connected to.

For instance, the example above would be provided in the following format:

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

Language Notes

C/C++ You must `#include "cave.h"`.

Pascal You must define the `unit Cave`, and you must also import the grader routines via `uses GraderHelpLib`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.