International Olympiad in Informatics 2013

July 2013 6-13

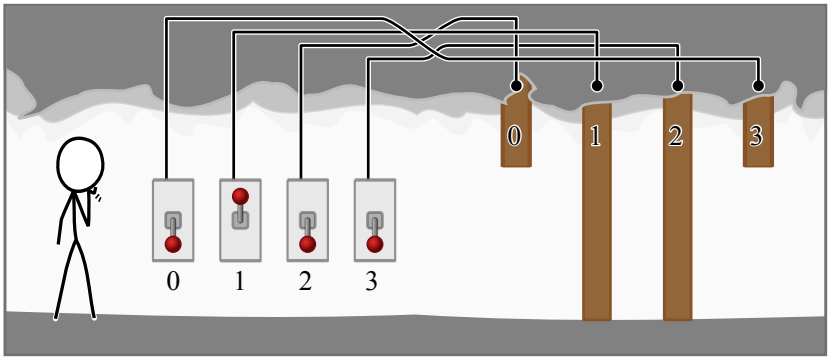
Brisbane, Australia

Day 2 tasks

cave

Hebrew — 1.0

הלכת לאיבוד באוניברסיטה והגעת למערה סודית. הכניסה למערה חסומה על-ידי מערכת של N דלתות שמסודרות אחת אחרי השניה. הדלתות נשלטות על-ידי N מתגים, שכל אחד מהם מחובר לדלת אחת בדיוק, כך שמתגים שונים מחוברים לדלתות שונות.



הדלתות ממוספרות 0, 1, ..., (N - 1) לפי הסדר, כאשר דלת מספר 0 היא הקרובה ביותר אליך. המתגים ממוספרים גם הם 0, 1, ..., (N - 1), אבל אתה לא יודע איזה מתג מחובר לאיזו דלת.

המתגים נמצאים כולם בכניסה למערה. לכל מתג יש שני כיוונים אפשריים: כיוון "למעלה" וכיוון "למטה". לכל מתג, אחד מהכיוונים הוא הכיוון הטוב והשני הוא הכיוון הרע. לכל מתג, הדלת שלו פתוחה אם ורק אם הוא בכיוון טוב. אתה לא יודע מה הכיוון הטוב ומה הכיוון הרע של כל מתג. ייתכן שבמתג אחד הכיוון הטוב הוא למעלה, ובמתג אחר הכיוון הטוב הוא דווקא למטה.

אתה רוצה להבין את מערכת האבטחה. כלומר, לגלות איזה מתג שולט בכל דלת ומה הכיוון הטוב של כל מתג. כדי לעשות זאת, אתה יכול לקבוע קונפיגורציה לבחירתך של כיווני המתגים, כלומר, אתה יכול לשים כל מתג בכיוון למעלה או למטה. לאחר שקבעת קומבינציה כזו, אתה מסתכל לכיוון הדלתות ומגלה מי היא הדלת הסגורה שהכי קרובה אליך. אתה לא רואה את שאר הדלתות הסגורות, כי היא מסתירה אותן.

אתם יכולים לנסות לכל היותר 70,000 קומבינציות של כיווני המתגים. עליכם לגלות לכל מתג לאיזו דלת הוא מחובר ומה הכיוון הטוב שלו.

פרטי מימוש

עליכם לממש אך ורק את הפונקציה exploreCave() . אתם יכולים לקרוא לכל היותר 70,000 פעמים לפונקציה tryCombination() . בסוף, אתם צריכים לקרוא לפונקציה answer() . הפונקציות האלה מתוארת להלן:

Grader Function: `tryCombination()`

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

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

Description

מערכת הבדיקה מספקת לכם את הפונקציה הזאת. זאת הפונקציה שמאפשרת לנסות קומבינציה לבחירתכם של כיווני המתגים. הפונקציה מחזירה את המספר של הדלת הסגורה הראשונה. אם כל הדלתות פתוחות, הפונקציה מחזירה -1 . הפונקציה רצה בזמן O(N) .

מותר לקרוא לפונקציה הזו לכל היותר 70,000 פעמים.

Parameters

- S : מערך בגודל N שמתאר את הכיוון שאתם בוחרים לכל מתג. אם האיבר S[i] הוא 0 זה אומר שאתם בוחרים לשים את מתג i בכיוון למעלה. אם S[i] הוא 1 זה אומר שאתם בוחרים לשים את מתג i בכיוון למטה.
- הפונקציה מחזירה את המספר של הדלת הסגורה הראשונה. אם כל הדלתות פתוחות, הפונקציה מחזירה -1 .

Grader Procedure: `answer()`

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

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

Description

עליכם לקרוא לפונקציה הזו לאחר שאתם מגלים מה הכיוון הטוב של כל מתג ולאיזו דלת כל מתג מחובר.

Parameters

- `S`: מערך בגודל `N` שמתאר את הכיוון הטוב של כל מתג. כלומר, זה בדיוק המערך שצריך להכניס לפונקציה `tryCombination()` כדי לפתוח את כל הדלתות.
- `D`: מערך בגודל `N` שמתאר לאיזו דלת מחובר כל מתג. כלומר, האיבר `D[i]` צריך להיות שווה למספר של הדלת שאליה מחובר מתג מספר i.
- הפונקציה הזאת לא מחזירה כלום. קריאה לפונקציה הזו תגרום לתוכנית להסתיים.

Your Procedure: `exploreCave()`

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

Pascal
```
procedure exploreCave(N: longint);
```

Description

זאת הפונקציה שעליכם לממש.

הפונקציה הזאת צריכה לגלות את הכיוון הטוב של כל מתג ולאיזו דלת מחובר כל מתג. היא עושה זאת על-ידי קריאות לפונקציה `tryCombination()`. לבסוף, היא קוראת לפונקציה `answer()` כדי לדווח את התשובה.

Parameters

- `N`: מספר הדלתות במערה.

Sample Session

נניח שהדלתות והמתגים הם כמו בתמונה.

Function Call	Returns	Explanation
`tryCombination([1, 0, 1, 1])`	`1`	זה מתאים למצב המתגים כמו שרואים בתמונה. כלומר, מתגים 0,2,3 למטה ומתג 1 למעלה. הפונקציה מחזירה 1 כי דלת מספר 1 היא הראשונה שסגורה.
`tryCombination([0, 1, 1, 0])`	`3`	Doors 0, 1 and 2 are all opened, while door 3 is closed.
`tryCombination([1, 1, 1, 0])`	`-1`	Moving switch 0 down causes all doors to be opened, indicated by the return value of `-1`.
`answer([1, 1, 1, 0], [3, 1, 0, 2])`	*(Program exits)*	אנחנו מדווחים שהקומבינציה הנכונה היא `[1,1,1,0]`, ושהמתגים `0,1,2,3` מחוברים לדלתות `3,1,0,2` בהתאמה.

Constraints

- Time limit: 2 seconds
- Memory limit: 32 MiB
- `1 ≤ N ≤ 5,000`

Subtasks

Subtask	Points	Additional Input Constraints
1	12	לכל `i`, מתג `i` מחובר לדלת `i`. נותר לגלות את הקומבינציה הנכונה.
2	13	מובטח שהקומבינציה הנכונה היא `[0, 0, 0, ..., 0]`. נותר לגלות לכל מתג - לאיזו דלת הוא מחובר.
3	21	`N ≤ 100`
4	30	`N ≤ 2,000`
5	24	*(None)*

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.