



International Olympiad in Informatics 2013

6-13 July 2013

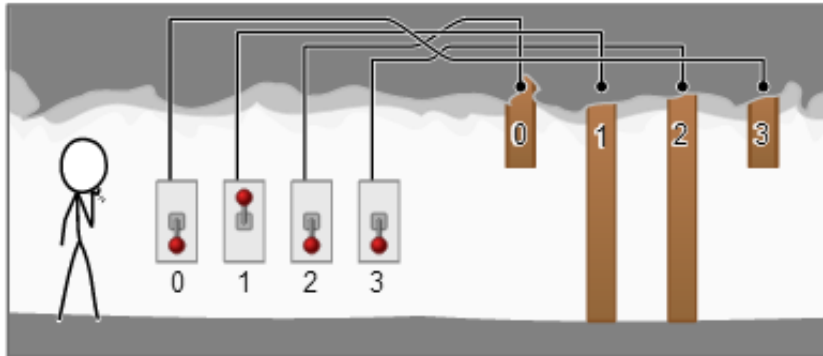
Brisbane, Australia

Day 2 tasks

cave

English — 1.0

სეირნობის დროს თქვენ დაიკარგეთ UQ ცენტრის მახლობლად და აღმოჩნდით საიდუმლო გამოქვაბულთან. მისი შესასვლელი დაბლოკილია დაცვის სისტემით, რომელიც შედგება მიმდევრობით განლაგებული N ცალი კარისაგან (კარები განლაგებულია არა ერთმანეთის გვერდი-გვერდ, არამეტ ერთმანეთის უკან). დაცვის სისტემას აგრეთვე გააჩნია N ცალი გადამრთველი, რომელთაგან თითოეული განსხვავებული მათგანი დაკავშირებულია განსხვავებულ კართან.



კარები გადანომრილია თანმიმდევრულად და ზრდადობით $0, 1, \dots, (N - 1)$ ისე, რომ 0 ნომრის მქონე კარი თქვენგან უახლოესია. გადამრთველები აგრეთვე გადანომრილია თანმიმდევრულად $0, 1, \dots, (N - 1)$. თუმცა, თქვენთვის უცნობია, თუ რომელი გადამრთველი რომელ კართან არის დაკავშირებული.

გადამრთველები განლაგებულია გამოქვაბულის დასაწყისში. ყოველი მათგანი ზედა ან ქვედა მდგომარეობაშია. თითოეული გადამრთველისთვის მხოლოდ ერთერთ პოზიციაში დაფიქსირება იწვევს შესაბამისი კარის გაღებას (ამ პოზიციას კორექტული ვუწოდოთ), ხოლო გადამრთველის საწინააღმდეგო მდგომარეობა იმავე კარს ხურავს.

თქვენ გსურთ შეისწავლოთ დაცვის სისტემა. ამისთვის შეგიძლიათ შეარჩიოთ გადამრთველთა მდგომარეობების ნებისმიერი კომბინაცია და შემდეგ მიხვიდეთ პირველივე დახურულ კარამდე. არცერთი კარი არ არის გამჭვირვალე, ამიტომ პირველივე დახურული კარის მიღმა ვერაფერს ხედავთ.

თქვენ გეძლევათ საშუალება, მოსინჯოთ გადამრთველების არაუმეტეს 70,000 კომბინაცია. თქვენი ამოცანაა განსაზღვროთ გადამრთველთა კორექტული პოზიციები ისე, რომ გააღოთ ყველა კარი. აგრეთვე უნდა განსაზღვროთ, რომელი გადამრთველი რომელ კართანაა დაკავშირებული.

იმპლემენტაცია

თქვენ უნდა გააგზავნოთ ფაილი, რომელიც მოახდენს `exploreCave()` პროცედურის იმპლემენტაციას, სადაც შეგიძლიათ მოახდინოთ `tryCombination()` ფუნქციის არაუმეტეს 70,000 გამოძახება. ამ პროგრამაში მუშაობა უნდა დაამთავროთ `answer()` პროცედურის გამოძახებით.

გრადერის ფუნქცია: `tryCombination()`

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

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

აღწერა

ეს ფუნქცია იმართება გრადერის მიერ. ის იძლევა საშუალებას შევამოწმოთ გადამრთველთა გარკვეული კომბინაცია და გავიგოთ პირველივე დახურული კარის ნომერი ამ კომბინაციისათვის. თუკი ყველა კარი გაღებულია, ფუნქცია აბრუნებს `-1`-ს. ეს ფუნქცია მუშაობს $O(N)$ დროში, ანუ შესრულების დრო უარეს შემთხვევაში N -ის პროპორციულია..

ფუნქცია გამოძახებულ უნდა იქნას არა უმეტეს 70,000 -ჯერ.

პარამეტრები

- `S`: N სიგრძის მასივი, რომელიც განსაზღვრავს ყველა გადამრთველის მდგომარეობას. `S[i]` ელემენტი შეესაბამება i -ურ გადამრთველს. მნიშვნელობა `0` აღნიშნავს გადამრთველის ზედა მდგომარეობას, ხოლო მნიშვნელობა `1` აღნიშნავს გადამრთველის ქვედა მდგომარეობას.
- *Returns*: პირველივე დახურული კარის ნომერი, ან `-1`, თუ ყველა კარი ღიაა.

გრადერის პროცედურა: **answer()**

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

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

აღწერა

ეს პროცედურა გამოიძახება მაშინ, როცა თქვენ დაადგენთ გადამრთველების ისეთ კომბინაციას, რომელიც გააღებს ყველა კარს და განსაზღვრავთ, რომელი კარი რომელ გადამრთველთანაა დაკავშირებული.

პარამეტრები

- **S**: **N** სიგრძის მასივი, რომელიც განსაზღვრავს ყველა გადამრთველის მდგომარეობას. მისი ფორმატი ემთხვევა ზემოთ მოცემულ `tryCombination()` ფუნქციაში ამავე მასივის ფორმატს.
- **D**: **N** სიგრძის მასივი, რომელიც განსაზღვრავს თითოეულ გადამრთველთან მიერთებულ კარის ნომერს. სხვაგვარად რომ ვთქვათ, `D[i]` ელემენტი წარმოადგენს `i`-ურ გადამრთველთან მიერთებული კარის ნომერს.
- *Returns*: ეს პროცედურა არაფერს აბრუნებს, რადგან მისით ხდება პროგრამის დასრულება.

თქვენი პროცედურა: **exploreCave()**

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

Pascal `procedure exploreCave(N: longint);`

აღწერა

თქვენმა ამოხსნამ უნდა მოახდინოს ამ პროცედურის იმპლემენტაცია.

ამ ქვეპროგრამამ უნდა გამოიძახოს გრადერის ფუნქცია `tryCombination()`, რათა დაადგინოს გადამრთველების კორექტული კომბინაცია და რომელი გადამრთველი რომელ კართანაა დაკავშირებული, და როგორც კი დაადგენს, გამოიძახოს `answer()`, რომელსაც გადასცემს საჭირო ინფორმაციას.

პარამეტრები

- N : გადამრთველთა და კართა რაოდენობა გამოქვაბულში.

შეტანის მაგალითი

დავუშვათ, კარები და გადამრთველები განლაგებული არიან ზემოთ მოცემული ნახაზის შესაბამისად:

ფუნქციის გამოძახება	Returns	განმარტება
<code>tryCombination([1, 0, 1, 1])</code>	1	შეესაბამება მოცემულ ნახაზს. გადამრთველები 0, 2 და 3 ქვედა მდომარეობაშია, ხოლო 1 - ზედა მდგომარეობაში. ფუნქცია აბრუნებს 1 -ს, რაც ნიშნავს, რომ პირველივე დახურულ კარს ნომერი 1 კარი წარმოადგენს.
<code>tryCombination([0, 1, 1, 0])</code>	3	კარები 0, 1 და 2 ღიაა, ხოლო კარი 3 დახურულია.
<code>tryCombination([1, 1, 1, 0])</code>	-1	0 ნომრის მქონე გადამრთველის მდგომარეობის შეცვლა აღებს ყველა კარს, ამიტომ ფუნქციის მიერ დაბრუნებული მნიშვნელობაა -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	<i>(Program exits)</i>	ჩვენ ვთვლით, რომ კორიქტული კომბინაციაა <code>[1, 1, 1, 0]</code> და გადამრთველები 0, 1, 2 და 3 დაკავშირებულია შესაბამისად 3, 1, 0 და 2 კარებთან.

შეზღუდვები

- დროის ლიმიტი: 2 წამი
- მეხსიერების ლიმიტი: 32 MiB
- $1 \leq N \leq 5,000$

ქვეამოცანები

ქვეამოცანა	ქულები	შეტანის დამატებითი შეზღუდვები
1	12	ყოველი i -სთვის i -ური გადამრთველი შეერთებულია i -ურ კართან. თქვენმა პროგრამამ მხოლოდ უნდა დაადგინოს გადამრთველთა კორექტული კომბინაცია.
2	13	კორიქიკოული კომბინაცია ყოველთვის იქნება $[0, 0, 0, \dots, 0]$. თქვენი ამოცანაა მხოლოდ დაადგინოთ, რომელი გადამრთველი რომელ კართან არის დაკავშირებული.
3	21	$N \leq 100$
4	30	$N \leq 2,000$
5	24	(None)

ექსპერიმენტირება

გრადერი თქვენს კომპიუტერში წაიკითხავს შემოსატან მონაცემებს ფაილიდან `cave.in` შემდეგი ფორმატის თანახმად :

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

აქ `N` წარმოადგენს გადამრთველების და კარების რაოდენობას, `S[i]` არის i -ური გადამრთველის კორექტული მდგომარეობა, ხოლო `D[i]` არის იმ კარის ნომერი, რომელიც შეერთებულია i -ურ გადამრთველთან.

ზემოთ მოყვანილი მაგალითი წარმოდგენილი იქნება შემდეგი ფორმატით:

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

შენიშვნები პროგრამირებული ენებისათვის

C/C++ თქვენ უნდა გამოიყენოთ `#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`).

მაგალითებისათვის ამოხსნათა შაბლონები იხილეთ თქვენს კომპიუტერზე.