



International Olympiad in Informatics 2013

6-13 July 2013

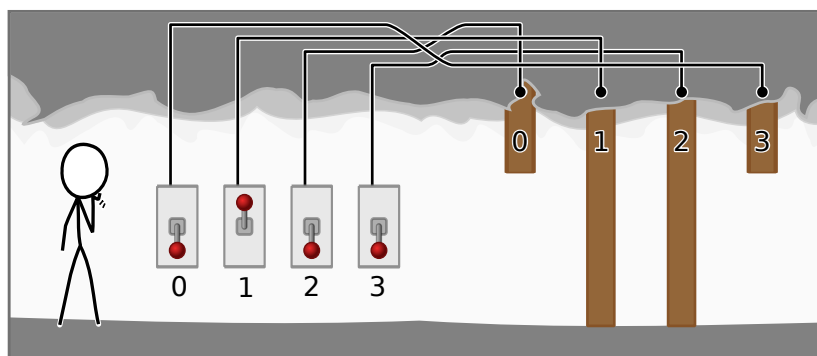
Brisbane, Australia

Day 2 tasks

cave

Polish — 1.0

W trakcie drogi z akademika do UQ Centre zauważyłeś wejście do sekretnej jaskini ukrytej głęboko pod kampusem. Wejście do jaskini prowadzi przez korytarz, w którym znajduje się N drzwi, umiejscowionych jedno za drugim. Zauważyłeś także układ N przełączników. Przełączniki te są połączone z drzwiami i służą do sterowania nimi. Każdy przełącznik jest podłączony do innych drzwi.



Drzwi są ponumerowane kolejno liczbami $0, 1, \dots, (N-1)$, przy czym drzwi numer 0 znajdują się najbliżej Ciebie. Przełączniki są również ponumerowane liczbami $0, 1, \dots, (N-1)$, jednak nie wiesz, którymi drzwiami steruje każdy z przełączników.

Przełączniki znajdują się przy wejściu do jaskini. Każdy przełącznik może być albo w pozycji w *górze*, albo w pozycji w *dół*. Tylko jedna z tych dwóch pozycji przełącznika powoduje, że podłączone do niego drzwi otwierają się. Jeśli przełącznik jest w drugiej pozycji, odpowiednie drzwi pozostają zamknięte. Pozycje otwierające drzwi mogą być różne dla różnych drzwi i, niestety, nie są Ci znane.

Chciałbyś rozkminić system bezpieczeństwa jaskini. W tym celu możesz dowolnie przestawiać przełączniki, a następnie wyruszyć korytarzem w kierunku jaskini, aby znaleźć pierwsze drzwi, które zostały zamknięte. Drzwi nie są przezroczyste, zatem dotarłszy do tych drzwi, nie widzisz żadnych z dalszych drzwi.

Twój wolny czas pozwala Ci na wypróbowanie co najwyżej 70,000 ustawień przełączników. Twoim zadaniem jest określenie, z którymi drzwiami jest połączony każdy z przełączników, jak również, w jakiej pozycji musi być on ustawiony, by otwierał drzwi.

Implementacja

Powinieneś zgłosić plik z implementacją procedury `exploreCave()`. Może ona wywoływać funkcję `tryCombination()` (co najwyżej 70 000 razy), a na koniec działania powinna wywołać procedurę `answer()`. Opis tych procedur podany jest poniżej.

Funkcja modułu oceniającego: `tryCombination()`

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

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

Opis

Funkcja ta zostanie zlinkowana z Twoim rozwiązaniem. Pozwala ona dowiedzieć się, dla podanego ustawienia przełączników, jaki jest numer pierwszych drzwi, które są zamknięte. Jeśli wszystkie drzwi są otwarte, funkcja zwraca `-1`. Funkcja działa w czasie $O(N)$.

Tę funkcję można wywołać co najwyżej 70 000 razy.

Argumenty

- `S`: Tablica długości `N` określająca pozycję każdego przełącznika. Wartość `S[i]` opisuje pozycję przełącznika `i`. Jeśli jest ona równa `0`, przełącznik `i` jest w pozycji do góry. Z kolei wartość `1` odpowiada pozycji w dół.
- *Wynik funkcji*: Numer pierwszych drzwi, które są zamknięte, lub `-1`, jeśli wszystkie drzwi są otwarte.

Procedura modułu oceniającego: `answer()`

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

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

Opis

Wywołaj tę funkcję po tym, jak odkryjesz ustawienie przełączników, które otwiera wszystkie drzwi, oraz układ połączeń między przełącznikami a drzwiami.

Argumenty

- **S**: Tablica rozmiaru **N** określająca ustawienie przełączników, które otwiera wszystkie drzwi. Jej format jest taki sam, jak (podany powyżej) format argumentu funkcji `tryCombination()`.
- **D**: Tablica rozmiaru **N** określająca, do których drzwi jest podłączony każdy z przełączników. Wartość `D[i]` powinna być równa numerowi drzwi połączonych z przełącznikiem `i`.
- *Wynik funkcji*: Funkcja ta nie zwraca żadnej wartości. Jej wywołanie powoduje zakończenie programu.

Twoja funkcja: `exploreCave()`

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

Pascal `procedure exploreCave(N: longint);`

Opis

Twoje rozwiązanie powinno zawierać implementację tej funkcji.

Funkcja ta powinna używać (dostarczonej) funkcji `tryCombination()`, aby wyznaczyć ustawienie przełączników otwierające wszystkie drzwi oraz układ połączeń między drzwiami a przełącznikami. Na końcu powinna wywołać funkcję `answer()`.

Argumenty

- **N**: Liczba przełączników oraz liczba drzwi w jaskini.

Przykład

Załóżmy, że drzwi i przełączniki są połączone tak, jak na zamieszczonym wcześniej obrazku.

Wywołanie	Wynik	Wyjaśnienie
<code>tryCombination([1, 0, 1, 1])</code>	1	Ten układ odpowiada rysunkowi. Przełączniki 0, 2 i 3 są w pozycji w dół, zaś przełącznik 1 - w pozycji w górę. Funkcja zwraca 1, co oznacza, że drzwi 1 to pierwsze (od lewej) zamknięte drzwi.
<code>tryCombination([0, 1, 1, 0])</code>	3	Drzwi 0, 1 i 2 są otwarte, zaś drzwi 3 są zamknięte.
<code>tryCombination([1, 1, 1, 0])</code>	-1	Przestawienie przełącznika 0 na pozycję w dół sprawia, że wszystkie drzwi są otwarte. Funkcja zwraca więc -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	<i>(Program kończy się)</i>	Orzekamy, że ustawienie <code>[1, 1, 1, 0]</code> otwiera wszystkie drzwi, a przełączniki 0, 1, 2 i 3 są połączone, odpowiednio, z drzwiami 3, 1, 0 i 2.

Ograniczenia

- **Maksymalny czas działania: 2 sekundy**
- **Limit pamięci: 32 MiB**
- $1 \leq N \leq 5\,000$

Podzadania

Podzadanie	Punkty	Dodatkowe ograniczenia
1	12	Dla każdego i , przełącznik i jest podłączony do drzwi i . Twoim zadaniem jest wyznaczenie ustawienia otwierającego drzwi.
2	13	Ustawienie <code>[0, 0, 0, ..., 0]</code> powoduje otwarcie wszystkich drzwi. Twoim zadaniem jest wyznaczenie połączeń między przełącznikami a drzwiami.
3	21	$N \leq 100$
4	30	$N \leq 2\,000$
5	24	<i>(brak)</i>

Uruchamianie lokalne

Przykładowy moduł oceniający na Twoim komputerze wczytuje dane z pliku `cave.in` w następującym formacie:

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

Wartość `N` oznacza tu liczbę drzwi i przełączników, `S[i]` to ustawienie przełącznika `i`, które otwiera drzwi, zaś `D[i]` to numer drzwi połączonych z przełącznikiem `i`.

Dane z powyższego przykładu powinny być więc podane w następującym formacie:

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

Uwagi natury językowej

C/C++ Użyj dyrektywy `#include "cave.h"`.

Pascal Zdefiniuj `unit Cave` oraz włącz funkcje modułu oceniającego przez dodanie `uses GraderHelpLib`. Wszystkie tablice są indeksowane od `0` (a nie od `1`).

Przykłady znajdziesz w przykładowych rozwiązaniach na Twoim komputerze.