Bartosz Walczak
Treść zadania, Opracowanie

Rafał Rusin
Program

Bramki

Dany jest układ złożony z n bramek. Bramki są ponumerowane od 0 do $n-1$. Każda bramka posiada pewną liczbę wejść i jedno wyjście. Wejścia i wyjścia mogą przyjmować stany 0, 1 lub $\frac{1}{2}$. Każde wejście jest połączone z dokładnie jednym wyjściem pewnej bramki. Stan wejścia jest równy stanowi wyjścia, z którym jest ono połączone. Każde wyjście może być połączone z dowolną liczbą wejść. Bramki o numerach 0 i 1 są specjalne — nie posiadają wejść i zawsze przyjmują określone stany na wyjściu: 0 dla bramki o numerze 0, 1 dla bramki o numerze 1.

Mówimy, że stan wyjścia bramki (krótko: stan bramki) jest **poprawny**, *jeżeli:*

a) *jest równy 0 i bramka ma więcej wejść w stanie 0 niż w stanie 1;*

b) *jest równy $\frac{1}{2}$ i bramka ma tyle samo wejść w stanie 0 co w stanie 1;*

c) *jest równy 1 i bramka ma więcej wejść w stanie 1 niż w stanie 0;*

d) *dana bramka jest specjalna, tzn. ma numer 0 lub 1, i jej stan jest równy odpowiednio 0 lub 1.*

Mówimy, że stan układu jest **poprawny**, *jeżeli stan każdej z bramek jest poprawny. Mówimy, że stan bramki jest* **zdeterminowany**, *jeżeli w każdym poprawnym stanie układu bramka ta przyjmuje ten sam stan.*

Zadanie

Napisz program, który:

- *Wczyta z wejścia opis układu bramek.*
- *Dla każdej bramki sprawdzi, czy jej stan jest zdeterminowany i jeżeli tak, to wyznaczy go.*
- *Wypisze wyznaczone stany bramek na wyjście.*

Wejście

Pierwszy wiersz standardowego wejścia zawiera liczbę bramek n, $2 \leqslant n \leqslant 10\,000$. Kolejne $n-2$ wierszy zawiera opisy połączeń bramek. Wiersz nr i opisuje połączenia łączące wyjścia bramek z wejściami bramki nr i. W wierszu tym znajduje się liczba k_i wejść bramki nr i, po której następuje k_i numerów bramek, $k_i \geqslant 1$. Są to numery bramek, których wyjścia są połączone z kolejnymi wejściami bramki nr i. Liczby w wierszach są pooddzielane pojedynczymi odstępami. Łączna liczba wszystkich wejść bramek nie przekracza $200\,000$.

Wyjście

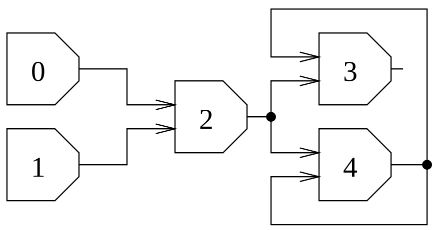
Twój program powinien wypisać na standardowe wyjście n wierszy. W zależności od stanu bramki numer $i-1$, i-ty wiersz powinien zawierać:

- `0` — *jeżeli stan bramki jest zdeterminowany i wynosi 0,*
- `1/2` — *jeżeli stan bramki jest zdeterminowany i wynosi $\frac{1}{2}$,*
- `1` — *jeżeli stan bramki jest zdeterminowany i wynosi 1,*
- `?` *(znak zapytania) — jeżeli stan bramki nie jest zdeterminowany.*

Przykład

Dla danych wejściowych:

```
5
2 0 1
2 4 2
2 2 4
```



poprawnym wynikiem jest:

```
0
1
1/2
?
?
```

Rozwiązanie

Aby uprościć opis, przyjmijmy, że w układzie nie ma bramek specjalnych, a niektóre wejścia mają po prostu ustalone stany (0 lub 1) i nie są połączone z wyjściami żadnych bramek.

Na bramki możemy patrzeć jak na przetworniki, które na podstawie stanów wejść obliczają stan wyjścia. Stan układu, który nie jest poprawny, możemy traktować jak stan, w którym jakaś bramka nie zdążyła jeszcze obliczyć poprawnego stanu wyjścia. W tym sensie układy w stanach niepoprawnych są niestabilne, a bramki dążą do ich stabilizacji.

Ogólny schemat rozwiązania można przedstawić następująco:

1. Ustaw stany wszystkich bramek na 0, po czym ustabilizuj układ.
2. Ustaw stany wszystkich bramek na 1, po czym ustabilizuj układ.
3. Stan bramki jest zdeterminowany wtedy i tylko wtedy, gdy jest taki sam po ustabilizowaniu w krokach 1 i 2.

Skąd wiadomo, że układ da się ustabilizować, czyli że istnieje stan poprawny? Wynika to z monotoniczności bramek. Ale na razie to tylko intuicja. W dalszej części postaramy się nieco sformalizować powyższe rozważania.

Stany układu będziemy oznaczać wielkimi, a bramki małymi literami. Stan bramki a, gdy układ jest w stanie S, będziemy oznaczać przez $s(a,S)$. *Wartością* bramki a w stanie układu S (oznaczenie $v(a,S)$) nazywamy liczbę:

- 0, gdy a ma więcej wejść w stanie 0 niż w stanie 1;
- $\frac{1}{2}$, gdy a ma tyle samo wejść w stanie 0, co w stanie 1;
- 1, gdy a ma więcej wejść w stanie 1 niż w stanie 0.

Stan bramki a jest poprawny wtedy i tylko wtedy, gdy $s(a,S) = v(a,S)$. Monotoniczność bramki oznacza, że jeżeli stan wejścia się zwiększa, to wartość bramki się nie zmniejsza, a jeżeli stan wejścia się zmniejsza, to wartość bramki się nie zwiększa.

Mówimy, że stan układu S jest *niski*, gdy dla każdej bramki a zachodzi $s(a,S) \leqslant v(a,S)$. Analogicznie, mówimy, że stan układu S jest *wysoki*, gdy dla każdej bramki a zachodzi $s(a,S) \geqslant v(a,S)$. Wprowadzamy również porównywanie stanów układu: jeżeli S_1 i S_2 są dwoma stanami układu, to piszemy $S_1 \leqslant S_2$, gdy dla każdej bramki a zachodzi $s(a,S_1) \leqslant s(a,S_2)$. Oczywiście nie każde dwa stany układu można ze sobą porównać.

Zdefiniujemy teraz operację poprawiania stanu bramki. Jeżeli stan bramki a w stanie układu S jest niepoprawny, to *poprawieniem* jej stanu nazywamy:

- zwiększenie go o $\frac{1}{2}$, gdy $s(a,S) < v(a,S)$;
- zmniejszenie go o $\frac{1}{2}$, gdy $s(a,S) > v(a,S)$.

Twierdzenie 1 *Dla każdego niskiego stanu układu S istnieje stan poprawny T taki, że jeżeli $S \leqslant P$, to $T \leqslant P$ dla dowolnego stanu poprawnego P.*

Dowód Niech S będzie dowolnym niskim stanem układu. Jeżeli S jest stanem poprawnym, to wystarczy przyjąć $T = S$ i teza jest oczywista. W przeciwnym przypadku istnieje bramka a, taka że $s(a,S) < v(a,S)$. Oznaczmy przez S' stan układu powstały z S przez poprawienie bramki a (stany pozostałych bramek się nie zmieniają). W wyniku operacji poprawiania wartości bramek mogą co najwyżej wzrosnąć. Zatem stan S' również jest stanem niskim.

Niech P będzie dowolnym takim stanem poprawnym układu, że $S \leqslant P$. Z monotoniczności bramek wynika, że dla dowolnej bramki b zachodzi $v(b,S) \leqslant v(b,P)$. W szczególności, dla poprawianej bramki a mamy $s(a,S) < v(a,S) \leqslant v(a,P) = s(a,P)$, skąd $s(a,S') \leqslant s(a,P)$. Stany pozostałych bramek się nie zmieniają, więc własność $s(b,S') \leqslant s(b,P)$ zostaje zachowana dla wszystkich bramek b, co oznacza, że $S' \leqslant P$.

Powyższą operację poprawiania bramek możemy powtarzać, otrzymując kolejno stany S', $(S')'$ itd., dopóki nie dostaniemy stanu poprawnego. Ponieważ w każdym kroku zwiększa się stan co najmniej jednej bramki oraz stan żadnej bramki się nie zmniejsza, takie postępowanie musi się skończyć, czyli otrzymamy jakiś stan poprawny T. Z zasady indukcji wynika, że otrzymany stan T spełnia tezę. ■

Twierdzenie 2 *Dla każdego wysokiego stanu układu S istnieje stan poprawny T, taki że jeżeli $P \leqslant S$, to $P \leqslant T$ dla dowolnego stanu poprawnego P.*

Dowód Analogiczny do dowodu twierdzenia 1. ■

Oznaczmy przez $\overline{0}$ stan układu, w którym wszystkie bramki są w stanie 0, a przez $\overline{1}$ stan układu, w którym wszystkie bramki są w stanie 1. Oczywiście $\overline{0}$ jest stanem niskim, a $\overline{1}$ jest stanem wysokim. Ponadto dla dowolnego stanu P (w szczególności, dla dowolnego stanu poprawnego) zachodzi $\overline{0} \leqslant P \leqslant \overline{1}$. Zatem na mocy twierdzeń 1 i 2 istnieją stany poprawne L i H, takie że $L \leqslant P \leqslant H$, dla dowolnego stanu poprawnego P. Jeżeli $s(a,L) < s(a,H)$, to stan bramki a nie jest zdeterminowany, gdyż jest różny w stanach poprawnych L i H. Natomiast jeżeli $s(a,L) = s(a,H)$, to dla dowolnego stanu poprawnego P zachodzi $s(a,L) = s(a,P) = s(a,H)$, więc stan bramki a jest zdeterminowany.

Rozwiązanie wzorcowe

Rozwiązanie wzorcowe znajduje stan L, implementując metodę przedstawioną w dowodzie twierdzenia 1 przy pomocy przeszukiwania wszerz:

1. Ustaw stany wszystkich bramek na 0.
2. Utwórz kolejkę Q bramek, których stany są niepoprawne (za niskie).
3. Dopóki kolejka Q nie jest pusta, powtarzaj następujące operacje:
   (a) Pobierz z kolejki Q dowolną bramkę a.
   (b) Popraw stan bramki a.
   (c) Wstaw do kolejki Q wszystkie bramki, których stany były poprawne, a w wyniku operacji poprawienia stanu a stały się niepoprawne.
   (d) Usuń z kolejki Q bramkę a, jeżeli po poprawieniu jej stan stał się poprawny (tylko stan bramki a mógł się stać poprawny).

Dla każdej bramki a pamiętana jest lista wszystkich bramek, których wejścia są połączone z wyjściem bramki a. Tylko te bramki mogą zmienić swoją wartość podczas poprawiania bramki a, a więc tylko je wystarczy rozpatrywać w punkcie (c) pętli. Cały układ jest więc reprezentowany przez graf, którego wierzchołki odpowiadają bramkom, a krawędzie połączeniom skierowanym od wyjścia do wejścia. Ponadto dla każdej bramki jest pamiętana liczba wejść w każdym z trzech dopuszczalnych stanów oraz bieżący stan bramki, co pozwala określić relację pomiędzy stanem bramki, a jej wartością w czasie jednostkowym. Stan każdej bramki jest poprawiany co najwyżej dwa razy. Zatem czas wykonania tej fazy rozwiązania wynosi $O(n+m)$, gdzie m jest łączną liczbą wszystkich połączeń między bramkami.

Analogicznie, w czasie $O(n+m)$ znajdowany jest stan H. Sprawdzenie dla każdej bramki a, czy $s(a,L) = s(a,H)$ oczywiście również można wykonać w czasie liniowym.

Powyższe rozwiązanie zostało zaimplementowane w `bra.cpp`.

Testy

Rozwiązania były oceniane na następującym zestawie testów:

- testy poprawnościowe: 1a, 1b, 2, 3a, 3b, 4;

- testy wydajnościowe: 5–11.

Testy 1a i 1b oraz 3a i 3b były zgrupowane w pary.

nr testu	n	m	opis
1a	2	0	przypadek szczególny $n = 2$
1b	7	9	
2	9	9	
3a	9	16	tylko stany niezdeterminowane[a]
3b	6	8	
4	27	32	
5	10000	10021	tylko stany zdeterminowane
6	1965	184365	prawie tylko stany zdeterminowane
7	1965	184365	prawie tylko stany zdeterminowane
8	9625	199769	prawie tylko stany zdeterminowane
9	9994	199904	tylko stany niezdeterminowane[a]
10	10000	199990	prawie tylko stany niezdeterminowane
11	10000	199990	prawie tylko stany niezdeterminowane

[a] Nie dotyczy bramek specjalnych