

Problema Scurtcircuit

Fișier de intrare stdin
Fișier de ieșire stdout

Un mare cercetător din orașul Zalău a plecat recent la studii în străinătate (cât încă mai are voie). Acesta a ales celebra University of Microcontrollers and Integrated Transistors și, prin urmare, de la începutul anului încoace visează și ziua și noaptea numai procesoare, circuite și porți logice.

Întors în orașul natal, cercetătorul s-a hotărât să vă dezvăluie cea mai interesantă problemă cu care se luptă în prezent:

Pentru N numere a câte K biți fiecare, să se construiască un circuit *suficient de scurt* care calculează suma acestor numere.

Definiția unui circuit

Un circuit este compus din următoarele elemente:

1. **Celule logice:** celule de memorie consecutive, numerotate de la 0 la $T + N \times K - 1$ care pot stoca exact câte un bit de informație, unde $N \times K$ este dimensiunea inputului, iar T este dimensiunea spațiului disponibil.
2. **Input:** un șir de biți așezați în primele celule logice. În cazul circuitului nostru de adunare, inputul o să se găsească în primele $N \times K$ celule. Atenție această secțiune de memorie nu poate fi *modificată*.
3. **Output:** un șir de biți așezați în ultimele celule logice. În cazul circuitului nostru de adunare, output-ul ar trebui să fie egal cu suma numerelor.
4. **Poartă logică:** o operație logică *binară* asociată unei celule. Această operație selectează alte două celule, calculează rezultatul operației logice dintre biții stocați în acele poziții și pune rezultatul în celula asociată ei. O operație logică este o funcție care are ca parametri 2 biți și returnează un bit.

Notă: fiecare celulă care nu face parte din input are *exact* o poartă logică asociată, dar o celulă poate fi folosită drept input de către mai multe porți logice.

Definim *adâncimea* unui circuit drept lungimea maximă a unui șir de porți logice astfel încât fiecare poartă logică depinde de rezultatul precedentei.

Putem descrie o poartă logică a printr-un șir de caractere s care conține caracterele 0, și 1 unde $a(i, j) = s(2 * i + j)$

Poziția în s	Valoarea
0	$a(0, 0)$
1	$a(0, 1)$
2	$a(1, 0)$
3	$a(1, 1)$

Tabela 1: Pozițiile șirului s și valorile asociate $a(i, j)$

Operator	Șir
(OR)	0111
^ (XOR)	0110
& (AND)	0001

Tabela 2: Operatori logici binari reprezentați ca șiruri

Definiția problemei

Pentru scopul acestei probleme, inputul va fi format din $N \times K$ biți reprezentând cele N numere, scrise în baza 2 și concatenate. Formal, celula $v[i \times K + j]$ = al j -lea bit (corespunzător puterii 2^j) al numărului i cu ($0 \leq i < N$, $0 \leq j < K$).

Valorile din următoarele T celule sunt calculate folosind porți logice. Formal, celula $v[N \times K + i]$ va fi calculată din cele două celule precedente.

Se dorește ca ultimele $K + \text{ceil}(\log_2(N))$ celule să ajungă să stocheze suma celor N numere, scrisă în baza 2 începând de la cel mai nesemnificativ bit. Formal suma cerută trebuie să fie $\sum_{0 \leq p} 2^p \times v[N \times K + T - K - \text{ceil}(\log_2(N)) + p]$, unde ceil reprezintă partea întreagă superioară a unui număr real.

Spunem că un circuit este suficient de scurt dacă adâncimea lui este mai mică sau egală decât T_{lim} , o valoare stabilită de comisie pentru fiecare test.

Date de intrare

Se dau numerele N și K .

Date de ieșire

Se va afișa T , numărul de celule folosite, urmat de T triplete de forma $a_i \ b_i \ c_i$ unde a_i este un șir binar de lungime 4, iar $\max(b_i, c_i) < N \times K + i$. Pentru ca soluția să fie considerată corectă, este necesar ca $K + \text{ceil}(\log_2(N)) \leq T \leq 3\,000\,000$.

Fiecare triplet are semnificația că funcția logică binară definită prin a_i este aplicată valorilor din celulele b_i și c_i (în această ordine) pentru a obține valoarea din celula $N \times K + i$.

Restricții

- $1 \leq N \times K \leq 100\,000$
- $K + \text{ceil}(\log_2(N)) \leq T \leq 3\,000\,000$

Vi se oferă un tabel cu testele care o să fie folosite la evaluare. Fiecare test valorează 5 puncte. Aceste puncte se vor acorda în funcție de parametrii $3\,000\,000 \geq D_{\text{lim}_1} \geq D_{\text{lim}_2} \geq D_{\text{lim}_3} \geq D_{\text{lim}_4} \geq D_{\text{lim}_5} \geq 1$.

Pentru fiecare test, fie D adâncimea circuitului construit de concurent. În cazul în care $D_{\text{lim}_1} < D$ atunci concurentul nu o să primească niciun punct. În cazul în care $D \leq D_{\text{lim}_5}$ atunci concurentul o să primească toate cele 5 puncte pe testul respectiv.

Fie i astfel încât $1 \leq i \leq 4$. Dacă $D_{\text{lim}_i} \geq D \geq D_{\text{lim}_{i+1}}$ atunci concurentul o să primească un punctaj de $\frac{(D_{\text{lim}_i} - D) \cdot (i+1) + (D - D_{\text{lim}_{i+1}}) \cdot i}{D_{\text{lim}_i} - D_{\text{lim}_{i+1}}}$ pentru acest test.

Puteți găsi mai jos un tabel care prezintă testele:

Exemple

stdin	stdout	Explicații
2 1	2 0110 0 1 0001 0 1	Acesta este un exemplu trivial în care adunăm două numere de câte un bit fiecare. Putem observa că dacă numerele sunt x și y , atunci rezultatul adunării este concatenarea binară a lui $x \text{ XOR } y$ și $x \text{ AND } y$.

Anexe

Vi se pune la dispoziție fișierul `check.cpp`. Odată ce este compilat și rulat, acesta citește numerele N și K din fișierul `input.in` și circuitul aferent lor din `circuit.out`. Apoi, această verifică

corectitudinea circuitului pentru 10 seturi de numere generate aleatoriu și de asemenea calculează adâncimea și numărul de porți logice.

Notă! Comportamentul fișierului de verificare pus la dispoziție poate fi diferit de comportamentul fișierului oficial de verificare.

Notă! Datorită volumului mare de date ce trebuie afișate, comisia științifică recomandă *puternic* utilizarea următoarelor 2 linii de cod:

```
ios_base::sync_with_stdio(0);
cin.tie(0);
```

Notă! Datorită volumului mare de date ce trebuie afișate, comisia științifică descurajează *puternic* folosirea `std::endl`, și recomandă *puternic* folosirea `'\n'`.

Teste

Există 26 de teste, fiecare în valoare de 5 puncte. Regăsiți în tabelul de mai jos valorile N și K și toate $D_{\lim_i}$ -urile folosite pentru calcularea punctajului.

Numărul Testului	N	K	$D_{\lim_1}$	$D_{\lim_2}$	$D_{\lim_3}$	$D_{\lim_4}$	$D_{\lim_5}$
1	2	2	300000	120	20	12	8
2	2	10	300000	167	67	16	12
3	2	100	300000	269	133	20	16
4	2	1000	300000	1236	200	28	24
5	2	10000	400000	10302	266	36	32
6	2	50000	2000000	50349	313	40	36
7	10	1	300000	124	24	20	16
8	100	1	300000	241	36	32	28
9	1000	1	300000	1208	52	48	44
10	10000	1	300000	10274	69	65	61
11	100000	1	2000000	100341	82	78	74
12	300	300	1800000	3030	1355	77	73
13	500	200	2000000	2433	1371	69	65
14	2000	50	2000000	2733	1238	77	73
15	5000	20	2000000	5493	1063	80	76
16	10000	10	2000000	10413	883	83	79
17	200	500	2000000	2433	1371	65	61
18	50	2000	2000000	2733	1238	60	56
19	20	5000	2000000	5493	1063	56	52
20	10	10000	2000000	10413	883	52	48
21	32	32	300000	600	500	45	41
22	16	64	300000	580	480	42	38
23	8	128	300000	520	420	38	34
24	64	16	300000	580	480	48	44
25	128	8	300000	520	420	49	45
26	10	10	300000	321	221	33	29