

anticip

Un sumator pe un bit este un mic dispozitiv cu 3 intrări și 2 ieșiri. El primește la intrare X_1 , X_2 și T_i . X_1 și X_2 sunt biți ce trebuie adunați, iar T_i este transportul anterior (ca intrare). La ieșire furnizează Y și T_o . Y este suma, iar T_o este transportul următor (ca ieșire).

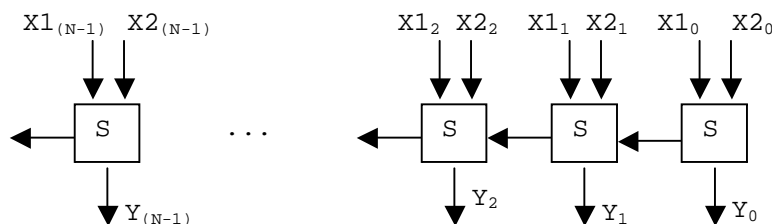
Pentru a formaliza aceste lucruri putem scrie

$$Y = (X_1 + X_2 + T_i) \bmod 2 \text{ (mod este restul împărțirii întregi)}$$

$$T_o = (X_1 + X_2 + T_i) \div 2 \text{ (div este câtul împărțirii întregi)}$$

Pentru a aduna numere pe N biți se folosesc N astfel de sumatoare. Ele sunt legate ca în

figura de mai jos, adică transportul de ieșire al unui sumator este transportul de intrare pentru următorul.



Problema cu aceste sumatoare pe mai mulți biți este că un sumator trebuie să aștepte transportul de la unitatea anterioară (exceptând primul sumator).

Dacă unui sumator pe un bit face calculul într-o secundă, atunci pentru un sumator pe N biți (format din N sumatoare pe un bit) vor fi necesare N secunde.

Pentru a îmbunătăți performanța acestor sumatoare pe N biți s-au introdus niște unități capabile să anticipeze transportul, adică intrarea T_i . Aceste unități verifică datele de intrare precedente $X_{1(i-1)}$ și $X_{2(i-1)}$. Dacă amândouă sunt 0 atunci T_i va fi 0, indiferent de ce primește acea unitate ca transport de intrare. De asemenea dacă amândouă sunt 1 atunci T_i va fi 1. Toate sumatoarele care folosind anticipația pot calcula transportul de la sumatorul precedent încep calculul odată cu primul sumator. Comunicarea transportului de la un sumator la următorul se realizează instantaneu.

Cercetătorii care au inventat aceste unități de transport vor să știe cu cât îmbunătățesc performanța sistemului și au hotărât să se facă toate adunările posibile, pentru a compara cu vechiul sistem. Prin toate adunările posibile se înțelege că se va aduna orice număr pe N biți cu orice număr pe N biți fix o dată (în total 4^N adunări). Ei vor să știe cât au durat aceste operații, adică suma tuturor timpilor pentru fiecare adunare.

Cerință

Scrieți un program care să determine numărul total de secunde necesare pentru toate adunările, folosind dispozitivele de anticipare.

Date de intrare

Fișierul de intrare `anticip.in` conține o singură linie pe care se află numărul natural N reprezentând numărul de biți.

Date de ieșire

Fișierul de ieșire `anticip.out` va conține o singură linie pe care va fi scris un număr natural reprezentând numărul de secunde necesare tuturor adunărilor posibile.

Restricții și precizări

- $0 < N < 51$
- Problema aceasta nu are nici o legătură cu vreun balaur.

Exemplu

anticip.in	anticip.out	Explicații
3	128	16 adunări se realizează într-o secundă; 32 adunări se realizează în două secunde; 16 adunări se realizează în trei secunde. De exemplu, adunarea 010+001 necesită 3 secunde. Adunarea 010+000 necesită 2 secunde. Adunarea 010+110 necesită o secundă (primul sumator adună biții din dreapta).

Timp maxim de execuție pe test: 0.5 secunde sub sistemul de operare Linux.

100 puncte

