

Restricția $N \leq 18$ ne permite să fixăm pentru care dintre cele N cifre se face trecerea peste ordine. Astfel, ajungem să aflăm N noi ecuații de tip $a_i + b_i = c'_i$ (în afară de cele M deja cunoscute).

Folosind cele $N + M$ ecuații putem crea un graf de $2 * N$ noduri (reprezentand cele N cifre din a și cele N cifre din b). Fiecare ecuație va reprezenta o muchie dintre 2 noduri.

Ideea de bază este de a reprezenta fiecare valoare din noduri sub forma $mX + n$, unde vrem să aflăm X . Pentru fiecare componentă conexă, luăm un nod și dăm valorile $m = 1$, $n = 0$ (deci valoarea din nod va fi egală cu X). Făcând parcurgerea DFS vom afla valorile lui m și n pentru fiecare nod. În arborele DFS, muchiile de întoarcere sunt cele care ne vor ajuta să rezolvăm ecuații. Dacă există o muchie de întoarcere între nodul i și j și avem valorile cunoscute m_i , n_i , m_j , n_j (va trebui să rezolvăm ecuația $m_i * X + n_i = m_j * X + n_j$). În cazul $m_i = m_j$, atunci dacă avem $n_i \neq n_j$ nu va exista soluție pentru X . Dacă $m_i \neq m_j$, atunci vom putea afla valoarea X , dacă nu a fost deja unic determinată (în cazul în care a fost deja determinată vrem să fie aceeași).

La final pentru fiecare componentă:

- X -ul rămâne o necunoscută (dar știm coeficientii)
- X -ul este unic determinat
- nu există soluție

Această idee este explorată și în problema [Graph](#) de la Baltica din 2020, care este explicată în detaliu [aici](#).

Ultimul pas, este să aflăm numărul de moduri în care putem da valori lui a și b .

Astfel, vom înmulți numărul de moduri de a da valori în fiecare componentă.

Pentru o componentă:

- dacă X -ul rămâne o necunoscută, vom calcula intervalul de valori ale lui X , astfel încât a_i și b_i sunt între 0 și $B-1$. Obs: valorile bune ale lui X vor fi un interval, deoarece reprezintă o intersecție de intervale. Numărul de moduri de a da valori în această componentă este dimensiunea intervalului.
- dacă X -ul este unic determinat, verificăm dacă toate valorile de a și b sunt între 0 și $B-1$. Numărul de moduri pentru componentă va fi 0 sau 1.
- Dacă nu există soluție numărul de moduri este 0.

Răspunsul final va fi suma numărului de moduri determinat de fiecare fixare a trecerii peste ordine.

Complexitate timp: $O(2^N * (N+M))$