Editorial RoAlgo PreOJI 2025

1-8 Martie 2025

Cuprins

1 Multumiri

Acest concurs nu ar fi putut avea loc fără următoarele persoane:

- Andrei Iorgulescu, Ștefan Dăscălescu, Traian Danciu, Vlad Munteanu, autorii și propunătorii problemelor și laureați la concursurile de informatică și membri activi ai comunității RoAlgo;
- Alex Vasiluță, fondatorul și dezvoltatorul principal al Kilonova;
- Ștefan Alecu, creatorul acestui șablon LaTeXpe care îl folosim;
- Andrei Chertes, David Curcă, Tudor Iacob, Susan, testerii concursului, care au dat numeroase sugestii și sfaturi utile pentru buna desfășurare a rundei;
- Ștefan Dăscălescu, Andrei Iorgulescu și Luca Mureșan, coordonatorii rundelor;
- Comunității RoAlgo, pentru participarea la acest concurs.

2 Balansoar

AUTOR: ANDREI PAUL IORGULESCU

2.1 Idee generală și soluție pentru $N = 4k$ și $N = 4k + 3$

Trebuie să atribuim fiecărui număr de la 1 la $2 \cdot N$ un coeficient întreg nenul între $-N$ și N, astfel încât să nu fie două numere cu același coeficient.

$$B = \sum_{i=1}^{2*N} (i * coef_i)$$

unde am notat cu B "balansul" șirului.
O idee de început este să grupăm numerele $2 \cdot i - 1$ și $2 \cdot i$ și să le dăm coeficienți de același modul, x și $-x$. Astfel, balansul va crește sau va scădea cu x. De fapt, dacă putem împărți șirul numerelor naturale de la 1 la N în două grupuri de sumă egală, putem crește B cu x-urile dintr-o grupă și îl scădem cu celelalte, obținând suma 0. Totuși, noi putem împărți numerele de la 1 la N în două grupuri cu sumă egală doar atunci când suma lor este divizibilă cu 2, adică $N \cdot (N + 1) = 4k$. Asta se întâmplă când $N = 4q$ și $N = 4q + 3$, adică pe aproximativ jumătate din teste. Un procedeu simplu de a obține una dintre cele două grupe este să pornim cu un $S = N \cdot (N + 1)/4$ și

să parcurgem numerele de la N la 1, mereu băgând numărul curent în grupă și scăzându-l din S dacă este mai mic sau egal cu S. Evident, celelalte numere formează cealaltă grupă.

2.2 Cazurile $N = 4k + 1$ si $N = 4k + 2$

Ne rămân cazurile când $N = 4k + 1$ și $N = 4k + 2$. Ne putem folosi totuși de ideea de a grupa numerele $2i - 1$ și $2i$. Să presupunem că avem o soluție cu $B = 0$ pentru un N, și vrem să atribuim coeficienții cu modul între $N + 1$ și $N + 4$ numerelor între $2 * N + 1$ și $2 * N + 8$ pentru a rămâne cu $B = 0$. Folosind cele 4 grupe cu diferența 1, putem pune numerele $N + 1$ și $N + 4$ cu plus, numerele $N + 2$ și $N + 3$ cu minus, astfel modificând B cu $N + 1 + N + 4 - (N + 2) - (N + 3) = 0$, deci lăsând $B = 0$. Așadar, dacă avem o soluție cu $B = 0$ pentru N, putem obține o soluție cu $B = 0$ pentru orice $N + 4k$.

2.3 N mic și finalizare

Cazurile $N = 1$ și $N = 2$ nu admit soluție cu $B = 0$, dar putem găsi soluții cu $B = 1$. Pentru $3 \leq N \leq 6$, există soluții cu $B = 0$, iar acestea pot să fie găsite fie pe foaie, de mână, fie cu ajutorul calculatorului. Astfel, pentru orice $N \geq 7$, putem găsi un $3 \leq n \leq 6$ astfel încât $N = n + 4k$, și putem aplica procedeul de mai sus pentru a produce o soluție optimă de la n la N.

2.4 Cod sursă

Soluție de 100

3 Aura

Autor: Stefan Dascalescu

3.1 Soluția oficială

La prima vedere, o soluție care vine ușor în minte este aceea că vom procesa actualizările brut, ținând minte suma elementelor după fiecare operație, dar această metodă va avea complexitatea $O(n \cdot q)$ ceea ce este mult prea încet.
Pentru a putea optimiza acest proces, trebuie să ne gândim la o metodă care să ne permită să putem efectua operațiile în care setăm toate valorile din șir rapid, păstrând în același timp și corectitudinea. O variantă simplă și ușor de folosit este aceea de a păstra folosind două variabile informații legate de ultima asemenea actualizare, astfel păstrând numărul folosit și indicele din șirul de actualizări la care s-a făcut acea operație.
Pentru operațiile de schimbare a unei valori, vom avea două cazuri: fie acea schimbare este prima corespunzătoare acelei poziții după ce am schimbat toate valorile din șir, fie nu este. În funcție de acest fapt, avem două cazuri care pot fi tratate foarte ușor folosind un vector de poziții, în care ținem evidența celor mai recente schimbări.
Astfel, complexitatea soluției devine $(n + q)$, ceea ce va obține punctajul maxim.

3.1.1 Cod sursă

Soluție de 100