Olimpiada Națională de Informatică, Etapa națională, Clasa a V-a Descrierea soluțiilor

Comisia științifică

23 martie 2026

Problema 1. Biatlon

Propusă de: prof. Nicoli Marius, Colegiul Național Frații Buzești, Craiova

Cerința 1.

Observăm că putem analiza problema independent pentru fiecare grup și însumăm rezultatele obținute. Loviturile care doboară o țintă le vom numi de tip 1 și loviturile care doboară două ținte deodată le vom numi lovituri de tip 2.

Ținând cont că dorim să executăm un număr maxim de lovituri și de faptul că la o lovitură de tip 2 se doboară două ținte deodată, rezultă că noi dorim să minimizăm numărul loviturilor de tip 2.

Intuitiv, dacă am avea 4 ținte alăturate, în loc să executăm o lovitură de tip 2 la primele două și o alta la ultimele două, observăm că este optim să executăm doar una, inițial, între țintele a doua și a treia, și apoi două lovituri de tip 1 la țintele rămase la margini, obținând un număr de trei lovituri. Putem dezvolta această idee astfel: la un șir mai lung de ținte alăturate, executăm prima lovitură la țintele 2 și 3 (lăsând ținta 1 izolată), apoi executăm o lovitură asupra țintelor 5 și 6 (lăsând ținta 4 izolată) și așa mai departe, deci, dacă am grupa țintele câte 3, reiese următoarea strategie: prima dată lovim o dată în a 2-a și a 3-a țintă din secvență, apoi rămâne izolată prima. Dacă numărul de ținte din grup nu este divizibil cu 3, în ultima secvență mai rămân una sau două ținte, care vor fi atinse deci dintr-o lovitură.

Fie g numărul de ținte dintr-un grup, avem:

- g este de forma $3k \implies 2k$ lovituri.
- g este de forma $3k + 1 \implies 2k + 1$ lovituri.
- g este de forma $3k + 2 \implies 2k + 1$ lovituri.

Se poate demonstra că strategia descrisă mai sus este una optimă.

Cerința 2.

Observăm că se dacă unim două grupuri, singura situație în care crește numărul total de lovituri ce se pot obține este când ambele grupuri au număr de ținte de forma $3k + 2$. Acest rezultat se deduce de exemplu analizând toate variantele de a împerechea două grupuri (cu toate cele 3 resturi posibile pentru un grup și pentru celălalt). Avem soluție deci doar dacă ultimul grup este de forma $3k + 2$. În acest caz putem afișa indicii valorilor de forma $3k + 2$ aflate între primele $n - 1$.

Problema poate fi abordată și simulând parcurgerea fiecărei secvențe de câte trei elemente și contorizarea a câte două lovituri (cu atenție la secvența incompletă de la final care se rezolvă dintr-o lovitură fie că rămâne de o țintă fie din două). Această implementare permite obținerea de punctaj parțial în funcție de tratarea corectă a cazurilor care apar.

Problema 2. Lacăte

Propusă de: prof. Lica Daniela, Centrul Județean de Excelență Prahova, Ploiești

Cerința 1.

Pentru fiecare din cele N coduri citite se va determina cifra maximă și cifra minimă. Dacă cele două au valori egale, cu alte cuvinte codul are toate cifrele egale, atunci se va număra ca amprentă formată din cifre identice.

Cerința 2.

Pentru obținerea unei amprente maxime vom șterge cifra minimă a codului. Dacă aceasta apărea de mai multe ori în codul inițial, atunci amprenta nouă va avea aceeași valoare cu cea inițială, în caz contrar amprenta nouă va avea o valoare mai mare. Se afișează suma amprentelor nou obținute.

Cerința 3.

Fiecărui cod îi vom asocia un număr format cu toate cifrele lui luate o singură dată, în ordine descrescătoare. De exemplu pentru codul 38804 vom asocia numărul 8430. Din enunț știm că $1 \leq$ codul oricărui lacăt din șir ≤ 100000 deci, numerele astfel asociate codurilor au cel mult cinci cifre.

Vom folosi un vector de frecvență $Marc$ pentru a marca, pentru fiecare număr din intervalul $[1, 99999]$, care este numărul codurilor care au acel număr asociat. Pentru exemplificare să presupunem că $Marc[82] = x$, adică sunt x coduri care sunt formate doar din cifrele 8 și 2. Cu ele se pot forma perechi echilibrate.

Să notăm 1, 2,.., x codurile, în ordinea crescătoare a indicilor. Codul notat cu 1 va crea $x - 1$ perechi echilibrate $(1, 2) \ldots (1, x)$. Codul notat cu 2 va crea $x - 2$ perechi echilibrate $(2, 3), (2, 4) \ldots (2, x)$, ș.a.m.d. Deci numărul total de perechi echilibrate se calculează cu folosind suma lui Gauss $(x - 1) * x/2$.

Numărul total de perechi echilibrate se obține ca sumă a numerelor de perechi echilibrate obținute pentru fiecare valoare asociată codurilor. Complexitatea algoritmului este $O(N)$.

O abordare care obține punctaj parțial are complexitate $O(N^2)$. Se determină după aceeași regulă numărul asociat fiecărui cod, care se rețin în ordine într-un vector de lungime N. Pentru fiecare pereche de indici se verifică egalitatea celor două numere asociate codurilor corespunzătoare și se contorizează egalitățile.

Problema 3. Șotron

Propusă de: Programator/specialist în domeniu, Măgureanu Livia, JetBrains, București

Există două cazuri în care pentru o aruncare unde piatra aterizează pe poziția x, aceasta să nu poată fi recuperată:

1. Dacă poziția x este deja blocată.

2. Dacă există un grup de 3 pătrățele consecutive blocate la stânga lui x.

Primul caz poate fi verificat folosind un vector caracteristic în care să marcăm pozițiile blocate.

Pentru al doilea caz, tot folosindu-ne de vectorul caracteristic menționat, dintre grupările de trei pătrățele consecutive blocate, o vom reține pe cea mai din stânga, adică cea care începe pe poziția cea mai mică.

Prin urmare, pentru fiecare valoare x citită, vom marca poziția x ca fiind blocată și vom verifica dacă se formează o nouă grupă de trei poziții blocate consecutive. Grupele pe care le avem de testat sunt $\{x-2, x-1, x\}$, $\{x-1, x, x+1\}$ și $\{x, x+1, x+2\}$, în măsura în care acestea există. Atunci când întâlnim o grupă nouă, comparăm poziția ei de început cu un minim și actualizăm dacă este cazul.

Pentru a verifica dacă poziția x este accesibilă trebuie ca ea să nu fie deja blocată și ca minimul pe care l-am calculat până la momentul respectiv să fie mai mare decât x.

Complexitatea timp a acestei soluții este $O(N+Q)$ și complexitatea de memorie este $O(N)$.

Echipa

Problemele pentru această etapă au fost pregătite de:

- Coman Isabela Patricia, Colegiul Național de Informatică Tudor Vianu, București, profesor
- Crantea Antonio-Cristian, Liceul Teoretic George Călinescu, București, profesor
- Iordaiche Eugenia Cristiana, Liceul Teoretic Grigore Moisil, Timișoara, profesor
- Lica Daniela, Centrul Judeţean de Excelenţă Prahova, Ploieşti, profesor
- Măgureanu Livia, JetBrains, București, Programator/specialist în domeniu
- Nicoli Marius, Colegiul Național Frații Buzești, Craiova, profesor
- Onuț Andrei, Yale University, New Haven, SUA, student
- Pintescu Alina, Colegiul Național Gheorghe Șincai, Baia Mare, profesor
- Popa Bogdan-Ioan, Liceul Teoretic Just 4Kids, București, profesor
- Tîmplaru Roxana Gabriela, Colegiul Ștefan Odobleja, Craiova, profesor