OLIMPIADA NAȚIONALĂ DE INFORMATICĂ PENTRU GIMNAZIU, ETAPA NAȚIONALĂ
CLASA A VIII-A
DESCRIEREA SOLUȚIILOR

COMISIA ȘTIINȚIFICĂ

Problema 1: Geologie

Propusă de: Stud. Livia Măgureanu,
Facultatea de Limbi şi Literaturi Străine,
Universitatea Bucureşti

Subtask 1: Pentru primul subtask, când toate numerele sunt prime, problema devine să determinăm pentru fiecare element al câtelea număr prim este și, apoi, să calculăm subsecvența de lungime maximă formată din numere prime consecutive.

Vom folosi Ciurul lui Eratostene pentru a determina numerele prime și numărul de ordine al acestora în lista numerelor prime. Odată ce avem această precalculare facută, putem găsi subsecvența de lungime maximă cu o parcurgere și un contor pe care îl resetăm de fiecare dată când întâlnim două numere alăturate care nu sunt prime consecutive. Complexitatea acestei soluții este $O(V_{max} \cdot log(V_{max}) + N)$, unde V_{max} este valoarea maximă din șir.

Subtask 2: Soluția se bazează pe cea de la subtask-ul anterior. Vom folosi în continuare Ciurul lui Eratostene pentru a determina numerele prime și pentru a descompune numerele din șirul dat în factori primi.

Se observă că fiecare număr din șir poate avea maxim 7 divizori primi diferiți. Vom considera, pe rând, fiecare poziție de început și fiecare număr prim cu care poate începe subsecvența pe care o căutăm. Astfel avem aproximativ $7 \cdot N$ variante de început posibile și pentru fiecare dintre ele vom calcula lungimea maximă a subsecvenței care pleacă din acel punct. Acest lucru se poate face parcurgând șirul în paralel cu șirul numerelor prime și verificând dacă numărul din șir la care am ajuns este divizibil cu numărul prim la care am ajuns.

Complexitatea finala a acestei abordări este $O(V_{max} \cdot log(V_{max}) + N^2)$ sau $O(V_{max} \cdot log(V_{max}) + N^2 \cdot log(V_{max})$ în funcție de implementarea folosită pentru descompunerea în factori primi.

Subtask 3 & 4: Soluția pentru ultimele două subtask-uri se bazează tot pe soluția pentru primul subtask. Diferența este că, pentru că acum numerele pot avea mai mulți divizori primi, vom calcula și reține soluția optimă pentru fiecare număr prim din descompunere în paralel.

Se observă că fiecare număr din șir poate avea maxim 7 divizori primi diferiți. Prin urmare calcula pentru fiecare poziție din șir care sunt subsecvențele care se pot termina pe poziția respectivă și lungimile acestora. Să presupunem că avem un șir în care al $i-1$-lea număr din șir este 31, al i-lea număr din șir este $84 = 2^2 \cdot 3 \cdot 7$ și al $i+1$-lea număr este $231 = 3 \cdot 7 \cdot 11$. Când ajungem pe poziția $i+1$ vom prelungi două dintre subsecvențele care se terminau pe poziția i, cele care corespund numerelor prime 2 și, respectiv, 7. Astfel pentru poziția $i+1$ vom avea 3 subsecvențe care se pot termina pe acolo: cea pentru 3 de lungime 2, cea pentru 3 de lungime 1 și cea pentru 11 de lungime 2. Putem implementa trecerea de la i la i fie asemănător cu o interclasare, fie folosindu-ne de lista numerelor prime pe care am construit-o în precalculare.

Complexitatea acestei soluții și punctajul obținut diferă în funcție de cum se implementează descompunerea în factori primi, dar soluția oficială are complexitatea $O(V_{max} \cdot log(V_{max}) + N)$.

Problema 2: Kmajo

Propusă de: Stud. Bogdan-Ioan Popa,
Facultatea de Matematică și Informatică,
Universitatea din București

Soluția de complexitate $O(N \cdot K)$ obține aproximativ 60 puncte. Se observă că este suficient să testăm subsecvențe cu lungimi cuprinse în intervalul $[K, 2 \cdot K]$ întrucât dacă există o subsecvență de lungime $L > 2 \cdot K$ care admite element majoritar, atunci cel puțin una dintre jumătățile sale va avea același element majoritar. Pentru fiecare L din intervalul $[K, 2 \cdot K]$ vom glisa o fereastră de lungime L peste întregul vector, menținând frecvențele tuturor elementelor. Atunci când găsim un element cu frecvența mai mare decât $[L/2]$ vom marca valoarea respectivă ca fiind parte din soluție.

Soluția de complexitate $O(N \cdot \sqrt{N})$ obține aproximativ 79 puncte. Dacă $K < 2 \cdot \sqrt{N}$, vom aplica soluția în $O(N \cdot K)$. Altfel se observă că vor fi maximum $O(\sqrt{N})$ candidați la a fi parte din soluție. Fiecare candidat va fi testat în $O(N)$. Pentru un candidat x fixat, vom construi un vector auxiliar B în care $B[i] = 1$ dacă $A[i] = x$ sau $B[i] = -1$ dacă $A[i] \neq x$. Tot ce rămâne de făcut este să găsim o subsecvență de lungime $\geq$ K care are suma > 0 în vectorul B.

Soluția de complexitate $O(N)$ obține 100 puncte. Vom testa fiecare valoare distinctă x dacă apare sau nu ca element majoritar într-o subsecvență de lungime cel puțin K. Fie $p_1 < p_2 < ... < p_T$ pozițiile pe care apare valoarea x în A, unde T reprezintă frecvența valorii x în A. Pentru ca x să fie element majoritar într-o subsecvență de lungime cel puțin K trebuie să existe doi indici $j \leq i$ astfel încât cel puțin una dintre cele două condiții să se respecte:

- $(i - j + 1) \cdot 2 > p_i - p_j + 1$, iar $p_i - p_j + 1 \geq K$
- $(i - j + 1) \cdot 2 > K$, iar $p_i - p_j + 1 < K$

Existența celor doi indici poate fi testată în $O(T)$ folosind tehnica Two-Pointers.

Problema 3: M and Ms

Propusă de: Stud. Mircea Măierean,
Facultatea de Matematică și Informatică,
Universitatea Babeș-Bolyai Cluj Napoca

Rezolvarea problemei se bazează pe un algoritm de tip greedy. Se observă faptul că, indiferent de numărul total de rânduri, o piramidă cu n rânduri va conține în totalitate $2^n - 1$ buline. Ca să minimizăm numărul total de piramide, încercăm la construcția fiecăreia să obținem o piramidă cu un număr cât mai mare de rânduri.

Cerința 1: Pentru subtaskul 1, calculăm numărul total de buline. La fiecare pas, selectăm cel mai apropiat număr de forma $2^p - 1$, care este mai mic sau egal cu totalul bulinelor. Scădem din total această valoare, și repetăm algoritmul până când numărul de buline devine 0. Răspunsul nostru este egal cu numărul total de repetări ale acestei proceduri.

La subtaskul 2, suma elementelor depășește limita maximă admisă de tipul de date unsigned long long, deci nu mai putem aplica direct procedeul descris la subtaskul 1. Pentru valorile maximale, suma totală a bulinelor nu depășește $n \cdot 2^{64} \approx 2^{21} \cdot 2^{64} = 2^{85}$. Ținem un vector de frecvență, care să poată reține frecvența fiecărei puteri. Pentru fiecare tip de buline existent, se descompune numărul în sumă de puteri ale lui doi, având termeni cât mai mari, și vom incrementa în vectorul de frecvență fiecare putere care apare în scrierea sumei (spre exemplu, $76 = 2^6 + 2^3 + 2^2$, deci vom incrementa vectorul pe pozițiile 2, 3, și 6). Pentru a putea forma o piramidă cu n rânduri, trebuie ca toate în vectorul de frecvență pozițiile de la 0 la $k1$ să fie nenule, unde k este cea mai mare putere care apare. Primul pas este să mutăm frecvențele pe poziții cât mai mari: dacă pe poziția i avem frecvența x, atunci putem incrementa pe poziția $i + 1$ valoarea $x/2$, iar elementul de pe poziția i va deveni restul împărțirii lui x la 2. Spre exemplu, dacă am avea frecvența poziției 2 egală cu 5 și frecvența poziției 3 egală cu 8, după transformare, vectorul va avea pe poziția 2 valoarea 1 și pe poziția 3 valoarea 10. La al doilea pas, iterăm de la 1 până la valoarea maximă găsită în descompunere. Pentru fiecare poziție i,

dacă valoarea frecvenței este nenulă, ne asigurăm că toate pozițiile cu un index având valoarea mai mică, sunt diferite de 0. Folosim proprietatea că $2^i = 2^{i-1} + 2^{i-1}$. Scădem o unitate de pe poziția i, și adăugăm 2 la poziția anterioară în vector, iar apoi decrementăm valoarea i. Repetăm acest lucru, până când valoarea frecvenței de pe poziția de la care am început acest procedeu devine 0. După ce ne-am asigurat că avem o secvență continuă de valori nenule, adăugăm la totalul piramidelor valoarea nenulă minimă care se găsește în vectorul de frecvență.

Cerința 2: Pentru rezolvarea subtaskului 1, procedeul este exact ca cel descris anterior, însă primul pas nu mai este necesar, ci putem să începem direct cu pasul al doilea, după ce am descompus numerele în sume de puteri ale lui 2.

Pentru rezolvarea subtaskului 2, translatarea pas cu pas a valorilor de la o poziție la alta nu va intra în timp. Pentru generarea optimă a piramidelor serioase, frecvența rândurilor este în ordine descrescătoare. Rândul 1 va apărea de cele mai multe ori, după aceea rândul 2, și tot așa, până la cel mai mare rând posibil. Obiectivul final este să prelucrăm tabelul de frecvențe, astfel încât să avem toate valorile în ordine descrescătoare. Translatăm valorile frecvențelor, începând de la cea mai mare poziție. Dacă valoarea de pe poziția i este mai mare decât valoarea poziției $i-1$, vom încerca să le apropiem cât de mult posibil. Fie x numărul minim de buline pe care trebuie să le scădem din frecvența de pe poziția i. Prin mutarea a x buline de pe poziția i, vor apărea $2 \cdot x$ buline pe poziția $i-1$.

Pentru a îl determina pe x, avem ecuația:

$fr[i-1] + 2 \cdot x \geq fr[i] - x, 1 < i \leq k$, unde fr este vectorul de frecvență, și k este cea mai mare putere care apare în scrierea unei sume ca puteri ale lui 2.

$fr[i-1] + 2 \cdot x \geq fr[i] - x \Leftrightarrow 3 \cdot x \geq fr[i] - fr[i-1] \Leftrightarrow x \geq \frac{(fr[i]-fr[i-1])}{3}$

Din valoarea frecvenței aflate la poziția i scădem x, și la poziția $i-1$ creștem valoarea cu $2 \cdot x$. Executăm acest procedeu pentru fiecare pereche de indici care respectă această proprietate.

Există riscul ca o apropiere să afecteze pozițiile mai mari, și să nu mai respecte condiția impusă. Se repetă procedeul descris până când tot vectorul conține toate elementele în ordine descrescătoare. La fiecare executare a procedeului, diferențele dintre frecvențe devin tot mai mici, așa că acesta se va repeta de puține ori.

Echipa

Problemele pentru această etapă au fost pregătite de:

- Prof. Flavius Boian, Colegiul Național „Spiru Haret", Târgu Jiu
- Stud. Livia Măgureanu , Facultatea de Limbi şi Literaturi Străine, Universitatea din Bucureşti
- Stud. Bogdan-Ioan Popa, Facultatea de Matematică și Informatică, Universitatea din București
- Stud. Mircea Măierean, Facultatea de Matematică și Informatică, Universitatea Babeș-Bolyai Cluj Napoca
- Prof. Isabela Patricia Coman, Colegiul Național de Informatică "Tudor Vianu", București
- Prof. Lucia Miron, Colegiul Naţional „Costache Negruzzi" Iaşi
- Prof. Stelian Ciurea, Universitatea "Lucian Blaga", Sibiu
- Prof. Dan Octavian Dumitrașcu, Colegiul Național "Dinicu Golescu", Câmpulung
- Prof. Daniel Popa, Liceul Teoretic "Aurel Vlaicu" Orăștie