OLIMPIADA NAȚIONALĂ DE INFORMATICĂ, ETAPA JUDEȚEANĂ
CLASA A VII-A
DESCRIEREA SOLUȚIILOR

COMISIA ȘTIINȚIFICĂ

Problema 1: Palindrom

Propusă de: Prof. Emanuela Cerchez, Colegiul Național "Emil Racoviță", Iași

Cerinţa 1. Vom citi succesiv numerele şi vom determina pentru fiecare număr citit numărul minim de cifre care trebuie să fie adăugate pentru a transforma numărul respectiv în palindrom. Pentru numere mari (de maximum 50 de cifre), vom citi fiecare număr caracter cu caracter, până la întâlnirea marcajului de sfârşit de linie şi vom reţine cifrele numărului într-un vector. Pentru punctaj parţial (numere de maximum 18 cifre) se va citi numărul într-o variabilă de tip *longlongint*, apoi se vor extrage cifrele numărului şi se vor plasa într-un vector. Pentru a determina numărul minim de cifre care trebuie adăugate la finalul numărului pentru a transforma acest număr în palindrom putem determina lungimea celui mai lung sufix al numărului care are proprietatea de a fi palindrom. Să notăm această lungime cu lgs, iar lungimea numărului cu lg. Numărul minim de cifre care trebuie să fie adăugate este $nr = lg - lgs$ (se adaugă la final primele nr cifre ale numărului în ordine inversată). Desigur, pentru punctaj parţial este posibilă şi o abordare "prin încercări":

- dacă numărul este deja palindrom, $nr = 0$;
- dacă numărul nu este palindrom, adăugăm la sfârşitul lui o cifră (prima cifră a numărului) şi verificăm dacă se obţine un palindrom (în acest caz nr va fi 1);
- apoi încercăm cu două cifre, trei cifre, ş.a.m.d.
- În cel mai defavorabil caz vom adăuga $nr = lg - 1$ cifre (primele $lg - 1$ cifre în ordine inversată).

Nu este necesar să reţinem toate numerele citite, vom reţine într-un vector nr cu n elemente numărul minim de cifre care trebuie să fie adăugate pentru a transforma fiecare număr din şir în palindrom. Rezultatul la cerinţa 1 este suma valorilor memorate în vectorul nr.

Cerinţa 2. Trebuie să determinăm cea mai lungă subsecvenţă a vectorului nr, construit la cerinţa anterioară, care are suma elementelor mai mică sau egală cu S. Pentru aceasta vom parcurge vectorul nr cât timp suma elementelor din secvenţa curentă (să o notăm sum) este $\leq S$. Când am ajuns la o poziţie i pentru care $sum + nr[i] > S$, elementul curent nu mai poate fi "înghiţit" în soluţie. Prin urmare:

- comparăm lungimea secvenţei curente cu lungimea maximă şi o reţinem dacă este mai mare;
- eliminăm elementele de la începutul secvenţei curente, actualizând corespunzător sum, până când este posibil să "înghiţim" valoarea $nr[i]$ în soluţie ($sum + nr[i] \leq S$).

Când parcurgerea vectorului nr s-a încheiat, trebuie să comparăm şi lungimea ultimei secvenţe cu lungimea maximă.
Pentru punctaje parţiale sunt posibile şi abordări de complexitate $O(n^2)$ sau $O(n^3)$.

Problema 2: Primprim

Propusă de: Stud. Ștefan-Cosmin Dăscălescu, Universitatea București

Pentru ambele cerințe va fi necesar să determinăm cât mai rapid pentru un număr dat distanța față de cel mai apropiat număr prim.
O primă abordare ar fi ca pentru fiecare număr să verificăm mai întâi dacă este prim (în acest caz, costul ar fi 0), iar în caz contrar ne deplasăm la stânga și la dreapta sa până când identificăm un număr prim, calculând apoi costul folosind formula dată din enunț. Totuși, o asemenea abordare ar avea complexitatea O($x * sqrt(valmax)$), unde x reprezintă distanța maximă față de un număr prim. Deoarece x este cel mult 57, o asemenea abordare nu obține punctaj maxim.
Pentru a optimiza această abordare, vom precalcula costurile pentru toate numerele de la 1 la 10^6. Pentru aceasta, vom utiliza ciurul lui Eratostene, pentru a genera numerele prime $\leq 10^6$, urmând ca mai apoi costul să fie calculat în O(1) pentru fiecare număr. Complexitatea precalculării este O($n \log \log n$).

Cerința 1. Vom citi succesiv numerele și vom afla succesiv costul pentru fiecare număr de la intrare, reținând într-o variabilă suma costurilor. În funcție de abordarea folosită pentru calcularea costurilor, se pot obține diverse punctaje parțiale, dar punctajul maxim pe cerință se poate obține doar folosind metoda bazată pe ciurul lui Eratostene, abordare explicată mai sus.

Cerința 2. Pentru această cerință vom citi succesiv operațiile și le vom executa.
Pentru a obține un cost total minim trebuie să adunăm cele mai mici p costuri. O abordare eficientă se bazează pe observația că pentru orice număr costul este cel mult 57, fapt ce ne permite să utilizăm un vector de frecvență fr, unde $fr[i]$ = numărul de elemente din vectorul v care au costul i.
Pentru fiecare operație, la modificarea unei valori din vector, vom decrementa frecvența costului pentru vechea valoare și vom incrementa frecvența costului pentru noua valoare. Pentru a determina costul total minim pentru a obține cel puțin p numere prime în vector (valoarea afișată după executarea operației), vom parcurge vectorul de frecvență de la stânga la dreapta ($i = 0, ..., 57$). La fiecare pas i, pentru a calcula costul total minim adunăm la o variabilă $cmin$ produsul dintre $fr[i]$ și i (există $fr[i]$ numere care pot fi transformate în numere prime cu costul i), iar într-o variabilă nr reținem câte numere prime au fost deja obținute. În momentul în care $nr + fr[i] > p$, parcurgerea se oprește și adunăm la $cmin$ doar costul obținerii celor $p - nr$ numere prime care mai sunt necesare (adică adunăm $fr[i] * (p - nr)$.

În funcție de cum selectăm cele mai mici p costuri și în funcție de cum calculăm costurile, se pot obține diverse punctaje parțiale.

Echipa

Problemele pentru această etapă au fost pregătite de:

- Prof. Cerchez Emanuela, Colegiul Național "Emil Racoviță", Iași
- Stud. Dăscălescu Ștefan-Cosmin, Universitatea București
- Prof. Frâncu Cristian, Clubul Nerdvana România, București
- Prof. Piț-Rada Ionel-Vasile, Colegiul Național Traian, Drobeta Turnu-Severin
- Prof. Panaete Adrian, Colegiul Național "A.T. Laurian", Botoșani
- Prof. Balașa Filonela, Colegiul Național "Grigore Moisil", București
- Prof. Moț Nistor, Școala "Dr. Luca", Brăila
- Prof. Boian Flavius, Colegiul Național "Spiru Haret", Târgu Jiu
- Stud. Pop Ioan-Cristian, Universitatea Politehnica București
- Prof. Dumitrașcu Dan Octavian, Colegiul Național "Dinicu Golescu", Câmpulung
- Stud. Tulba-Lecu Theodor-Gabriel, Universitatea Politehnica București