OLIMPIADA NAȚIONALĂ DE INFORMATICĂ
CLASA A VII-A
DESCRIEREA SOLUȚIILOR

COMISIA ȘTIINȚIFICĂ

Problema 1: Expresie

Propusă de: prof. Nistor Moț, Școala ”Dr. Luca” Brăila

Expresia va conține $p+q+1$ termeni, iar pentru a obţine rezultatul maxim va trebui să facem termenii care se adună cât mai mari, iar termenii care se scad cât mai mici.
Prima idee clară este că vom forma cei q termeni care se scad din câte o cifră. Dar termenii care se adună? Mai puţin intuitiv este faptul că şi la aceşti termeni vom urma aceeași idee: toţi vor avea o singură cifră, exceptând unul care va fi cel mai mare termen posibil. Pentru justificare analizăm câteva cazuri particulare:

- Dacă $p+q=n-1$ toţi termenii vor avea o singură cifră; vom pune semnul – în faţa celor mai mici q cifre (exceptând prima cifră) şi semnul + în faţa celorlalte cifre.
- Dacă $p=0$ și $q=1$, vom scădea ultima cifră din numărul format de primele $n-1$ cifre.
- Asemănător procedăm pentru toate cazurile când $p=0$: scădem ultimele q cifre din numărul format din primele $n-q$ cifre.
- Dacă avem $p=1$ și $q=0$, rezultatul maxim se obţine adunând prima sau ultima cifră la numărul format din celelalte cifre. Să justificăm pentru un număr de 6 cifre ($\overline{abcdef}$), dar raţionamentul este valabil pentru orice număr de cifre $sn>3$): $\overline{abcd}+\overline{ef}<\overline{abcde}+f$ (pentru că $\overline{abcd}+(10\cdot e+f)<10\cdot\overline{abcd}+e+f$, sau $9\cdot e<9\cdot\overline{abcd}$).
- Asemănător procedăm pentru toate cazurile când $q=0$; adunăm la cel mai mare număr format din $n-p$ cifre consecutive celelalte cifre.

Algoritmul general: căutăm cel mai mare număr format din $n-p-q$ cifre situate pe poziții consecutive din care scădem cele mai mici q cifre dintre cele rămase (excluzând-o pe prima) şi adunăm celelalte p cifre.
Dacă $n-p-q<18$ se pot face calcule cu numere întregi pe 64 biţi, dacă nu, trebuie implementate operaţiile de adunare şi scădere pentru numere mari, deoarece limbajul nu are (încă) un tip de date pentru numere cu până la 1000 cifre.

Problema 2: Pictura

Propusă de: prof. Emanuela Cerchez, Colegiul Național ”Emil Racoviță” Iași

Parcurgem matricea care reprezintă pictura şi identificăm elementele “vârf”. Pentru fiecare vârf identificat adăugăm o nouă “culoare pură” şi “vopsim“ pictura în noua culoare adăugată
Culorile pure sunt reprezentate printr-o singură valoare numerică, dar culorile obţinute prin amestecare sunt constituite dintr-o succesiune de valori numerice. Prin urmare vom reprezenta o culoare ca un vector în care reţinem toate valorile numerice corespunzătoare culorilor pure care intră în componenţa acesteia. Memoria disponibilă permite utilizarea vectorilor alocaţi static, însă utilizarea vectorilor alocaţi dinamic (clasa vector din STL) ar conduce la o soluţie cu un consum mic de memorie.

Pentru a vopsi pictura într-o nouă culoare c, începând din poziţia vârfului curent, vom utiliza o stivă S pe care o iniţializăm cu poziţia vârfului curent.

Cât timp stiva S nu este vidă:

- Extragem elementul din vârful stivei (să-l notăm p).
- Parcurgem vecinii elementului p şi verificăm pentru fiecare vecin dacă are o înălţime strict mai mică decât înălţimea poziţiei curente p (pentru a permite scurgerea culorii) şi dacă nu a fost deja vopsit în culoarea c (pentru a nu vopsi un pixel în aceeaşi culoare de mai multe ori).
- Pentru fiecare vecin care îndeplineşte condiţiile de mai sus, adăugăm culoarea c în lista culorilor corespunzătoare acestei poziţii din matrice şi adăugăm în stivă acest vecin.

Algoritmul de vopsire este un algoritm asemănător cu un algoritm de FILL, numai că este posibilă parcurgerea aceleiaşi poziţii de mai multe ori, pentru fiecare culoare pură adăugată.

Cerința 1: Pentru a rezolva cerinţa 1 este suficient să determinăm lungimea maximă a succesiunilor de culori obţinute după vopsire.

Cerința 2: Pentru a rezolva cerinţa 2 trebuie să identificăm numărul de culori distincte. Pentru aceasta putem sorta succesiunile de culori şi să numărăm apoi în timp liniar culorile distincte. Deoarece succesiunile de culori obţinute sunt deja ordonate crescător, compararea acestora din punct de vedere lexicografic este facilă.

Cerința 3: Pentru a rezolva cerinţa 3, trebuie să identificăm fiecare zonă formată din pixeli de aceeaşi culoare, să determinăm aria fiecărei zone şi apoi să determinăm maximul ariilor. Pentru a determina aria unei zone, aplicăm un algoritm de FILL.

Problema 3: Secv9

Propusă de: stud. Ioan-Cristian Pop, Universitatea Politehnica București

Observație inițială: cifra de control a unui număr nenul x se poate obține prin calcularea restului împărțirii lui x la 9, cu precizarea că, dacă restul este 0, atunci cifra de control este egală cu 9. Dacă x este egal cu 0, atunci cifra de control este egală cu 0.

De exemplu, pentru $x = 175$, cifra de control este egală cu 4. De asemenea, restul împărțirii la 9 este egal cu 4.

Sume parțiale: pentru rezolvarea celor 2 cerințe, ne vom folosi de sume parțiale ($sp[i]$ reprezintă suma tuturor numerelor de la x_1 până la x_i, și se poate calcula astfel: $sp[i] = sp[i-1] + x_i$).

Astfel, pentru doi indici i și j cu $i \leq j$, dacă $sp[j] - sp[i] > 0$ și $(sp[j] - sp[i])\%9 = 0$, atunci secvența $x_{i+1}, x_{i+2}, \ldots, x_j$ este de tip secv9.

Cerința 1: este suficient să căutăm, pentru fiecare rest r la împărțirea cu 9 de la 0 la 8, cea mai din stânga apariție st, respectiv cea mai din dreapta apariție dr a unei sume parțiale cu restul r din vectorul sp. Dacă suma $sp[dr] - sp[st] > 0$, atunci secvența $x_{st+1}, x_{st+2}, \ldots, x_{dr}$, este un candidat. Reținem o astfel de secvență de lungime maximă.

Cerința 2: construim un vector de frecvență f, unde $f[i]$ reprezintă câte sume parțiale au restul la împărțirea cu 9 egal cu i. Dacă avem $f[i]$ sume cu restul i, înseamnă ca vor exista $f[i] \times (f[i]-1)/2$ secvențe secv9 (grupând două câte două).

Vom număra numărul de secvențe pentru toate resturile de la 0 la 8 cu metoda precizată mai sus. Se observă însă, un caz particular: sumele cu restul 0 pot avea suma propriu-zisă egală cu 0, nefiind o secvență secv9. De aceea, este necesar să scădem din rezultat numărul de secvențe din șir cu suma egală cu 0.

Pentru a număra câte secvențe au suma egală cu 0, este suficient să identificam secvențele de 0 din șir. O secvență de k 0-uri consecutive conține $k \times (k-1)/2$ secvențe cu suma egală cu 0.

Echipa

Problemele pentru această etapă au fost pregătite de:

- Prof. Emanuela Cerchez, Colegiul Național ”Emil Racoviță” Iași
- Prof. Filonela Bălașa, Colegiul Național ”Grigore Moisil” București
- Prof. Florentina Ungureanu, Colegiul Național de Informatică Piatra Neamț
- Prof. Claudiu-Cristian Gorea-Zamfir, Liceul de Informatică ”Grigore Moisil” Iași
- Stud. Dumitru Ilie, Facultatea de Matematică-Informatică, Universitatea București
- Stud. Ioan-Cristian Pop, Universitatea Politehnica București
- Prof. Nistor Moț, Școala ”Dr. Luca” Brăila
- ing. Gabriel Tulbă-Lecu, Jane Street, UK
- lector dr. Paul Diac, Facultatea de Informatică, Universitatea ”Alexandru Ioan Cuza”, Iași