OLIMPIADA NAȚIONALĂ DE INFORMATICĂ
CLASA A VII-A
DESCRIEREA SOLUȚIILOR

COMISIA ȘTIINȚIFICĂ

Problema 1: Dominew

Propusă de: Prof. Flavius Boian, Colegiul Național "Spiru Haret", Târgu Jiu

La citire se reține elementele în 2 vectori distincți (primul va avea N elemente, iar cel de-al doilea M elemente).

În primul rând, se poate observa că numărul maxim de actualizări este mult mai mic decât numărul de valori inițiale ($M \leq 8000$), mult mai mic decât valoarea maximă a lui N, ceea ce ne duce cu gândul la a procesa separat cei doi vectori pentru a obține răspunsul într-un timp optim.

Pe măsură ce se citesc elementele din cel de-al doilea vector se identifică poziția pe care ar trebui inserate acestea în vectorul sortat până la acel moment. Vom nota valoarea pe care o verificăm la un pas cu x. Pentru fiecare element din șirul de M valori, va trebui mai întâi să căutăm binar în vectorul cu valorile inițiale poziția în care se află cea mai mare valoare mai mică decât x. Să notăm această poziție cu a. Apoi, vom afla și câte valori din vectorul de lungime M situate înaintea valorii curente sunt mai mici decât x. Datorită valorii mici a lui M, putem face acest lucru verificând fiecare poziție anterioară poziției curente. Să notăm acest răspuns cu b.

Astfel, răspunsul corespunzător pentru o valoare citită va fi $a + b + 1$.

Soluția are complexitatea totală $O(M^2 + M * \log N)$.

Soluție alternativă. Se vor citi mai întâi valorile și se vor sorta valorile din vectorul de lungime M. Apoi, vom aplica un algoritm de interclasare pentru a răspunde în manieră offline la toate întrebările, memorând mai întâi poziția unde se găseau acele actualizări în vectorul al doilea, având grijă să prioritizăm valoarea din vectorul al doilea în caz de egalitate, pentru a respecta restricția din enunț.

Această soluție are complexitatea totală $O(M^2 + M + N)$.

Problema 2: Pix

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

Reţinem într-un vector nr capacităţile distincte $\leq Vmax$.
Deoarece se garantează că există soluţie, obligatoriu $nr[1] = 1$.
Vom lua o cutie de tipul 1.
Analizăm apoi $nr[2]$. Dacă $nr[2] > nr[1]$ atunci comenzile pentru capacitatea x din mulțimea $\{1, 2, \ldots, nr[2] - 1\}$ nu pot fi obţinute decât din mai multe cutii cu capacitatea $nr[1] = 1$. Deci vom lua $nr[2] - 1$ cutii cu capacitatea $nr[1] = 1$.
Pentru a vedea câte cutii cu capacitatea $nr[2]$ luăm trebuie să analizăm $nr[3]$, pentru ca toate valorile până la $nr[3] - 1$ inclusiv să fi obţinute.
Observăm că dacă am lua j cutii de capacitate $nr[2]$ am obţine toate comenzile cu capacitatea x din mulțimea $\{1, 2, \cdots (j + 1) * nr[2] - 1\}$. Să generalizăm acum.

Să notăm cu *sum* suma capacităţilor cutiilor luate până la pasul i (acestea reprezintă valoarea maximă pentru care avem soluţie până la acest pas).
Câte cutii cu capacitatea $nr[i]$ vom lua?
Dacă $nr[i+1]-1 \leq sum$ nu are rost să luăm cutii cu capacitatea $nr[i]$.
Determinăm câte de cutii cu capacitatea $nr[i]$ sunt necesare astfel încât $nr[i+1]-1 \geq cate * nr[i] + sum$.
Pentru ca formula să funcţioneze şi pentru $i = n$ vom iniţializa componenta $n+1$ a vectorului nr cu $Vmax+1$, astfel că la pasul n vom asigura toate comenzile pentru capacităţile $x \leq Vmax$.

```
nr[n+1]=Vmax+1;
rez=nr[2]-1; sum=n[2]-1; //in rez contorizam cutiile
for (i=2; i<=n; i++)
     {
       if (nr[i+1]-1<=sum) continue;
       cate=(nr[i+1]-1-sum)/nr[i];
       if (cate*nr[i]+sum<nr[i+1]-1) cate++;
       rez+=cate;
       sum+=cate*nr[i];
     }
```

Problema 3: Secvmin

Propusă de: Stud. Theodor-Gabriel Tulba-Lecu și Ioan-Cristian Pop, Universitatea Politehnica București

Să considerăm cele mai mici valori distincte din șir $y_1, y_2, ..., y_k$.

Pentru fiecare poziție i din șir vom calcula și vom actualiza valorile $last_1, last_2, ..., last_k$ cu semnificația $last_j$ este cea mai mare poziție mai mică sau egală decât i în care a apărut valoarea y_j.

Se observă că o secvență care are capătul dreapta în poziția i va conține toate cele k valori minime dacă și numai dacă capătul stânga al secvenței este mai mic sau egal decât toate valorile $last_j$.

Pentru a obține soluția, la fiecare i actualizăm șirul *last* și adunăm la soluție valoarea minimă a acestui șir.

Pentru a forma și actualiza șirul *last* este necesar ca în prealabil să avem calculate cele k valori minime, dar aceasta se poate realiza încă de la citire prin inserarea în șirul y a celor mai mici k valori atunci când ele apar.

Șirul *last* se inițializează cu 0 și se actualizează de fiecare dată când la o poziție i apare una dintre cele k valori minime. Soluția are complexitatea $N * K$

Echipa

Problemele pentru această etapă au fost pregătite de:

- Prof. Cerchez Emanuela, Colegiul Național "Emil Racoviță", Iași
- Stud. Dăscălescu Ștefan-Cosmin, Universitatea București
- Prof. Pițˇ-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