DESCRIEREA SOLUTIILOR, OLIMPIADA NATIONALA DE INFORMATICA, CLASA A IX-A ETAPA JUDETEANA 13 MARTIE 2022

Problema 1: balba

Propusa de: student Gheorghe Liviu Armand

Cerința 1 - 40 de puncte

Pentru a rezolva această cerință, trebuie să facem anumite observații asupra modului în care un număr se schimbă după o bâlbâială.

Observăm că dacă în numărul X se află 2 cifre egale alăturate, se va obține același număr dacă cifra pe care o va repeta regele, adică cifra bâlbâită, este prima dintre acestea sau a 2-a dintre acestea. De asemenea, dacă în numărul X există 2 cifre egale care nu sunt alăturate, se vor obține numere diferite dacă cifra pe care o va repeta regele este prima dintre acestea sau a 2-a dintre acestea. Evident, numărul obținut prin bâlbâirea unei cifre va fi diferit de numărul obținut prin bâlbâirea unei cifre diferite ca valoare.

Astfel, deducem că prin bâlbâirea oricărei cifre dintr-o secventă de cifre egale din numărul X se va obține exact același număr, iar prin bâlbâirea unei cifre care nu este în acea secvență se va obține un număr diferit de cel obținut anterior.

Așadar, numărul de numere diferite pe care X le poate genera printr-o bâlbâiala este, de fapt, numărul de secvențe de cifre egale din numărul X.

Pentru a afla numărul de numere diferite care devin X după o bâlbâială, ne bazăm tot pe observațiile trecute și pe faptul că dacă o cifră a fost bâlbâită, atunci în numărul X aceasta se va afla într-o secvență de cifre egale formată din minim 2 cifre (cu alte cuvinte, cifra care a fost bâlbâită în numărul inițial nu se poate regăsi în X într-o secvență de cifre egale formată doar dintr-o cifra).

Deci, numărul de numere diferite care devin X după o bâlbâială este, de fapt, numărul de secvente de cifre egale de lungime minim 2 din numărul X.

O soluție a acestei cerințe are complexitatea timp $O(n)$.

Cerința 2 - 60 de puncte

Pentru rezolvarea acestei cerințe există mai multe abordări, iar în continuare vom prezenta 3 dintre aceste soluții care obțin punctajul maxim. Am notat cu Σ numărul total de cifre diferite care ar putea apărea în numărul X, întrucât problema s-ar putea generaliza, lucrând în acest fel cu un Σ mult mai mare. În cazul nostru $\Sigma = 10$.

Mai întâi vom pune cifrele numărului X într-un vector de frecvență pentru a putea lucra mai ușor și mai rapid cu ele.

Un prim aspect important este că un număr *palilindrom* este un palindrom fără o cifră(și anume cifra care se va adăuga după bâlbâială). Observăm că într-un palindrom, fiecare cifră apare de un număr par de ori, mai puțin eventuala cifră care se poate afla în mijloc dacă acesta are lungime impară. Așadar, într-un palindrom avem maxim o cifră de frecventă impară, iar restul cifrelor au frecvență pară. Astfel, deducem ca un *palilindrom* poate avea maxim 2 cifre de frecventă impară și restul de frecventă pară, întrucât bâlbâirea unei cifre va avea ca efect schimbarea parității frecvenței cifrei respective.

Deci, atenția noastră va pica asupra cifrei bâlbâite și a cifrei din mijloc.(Aveți grijă la cazurile când acestea au aceeași valoare sau coincid!). Dacă găsim un mod bun de a afla aceste cifre, atunci va trebui să vedem cum putem insera doar câte o apariție a fiecărei dintre aceste 2 cifre în cel mai mare palindrom care se poate forma cu restul cifrelor din număr. Cifra care se va bâlbâi și pe care noi o vom insera nu este fix cifra care se va bâlbâi, dar este cifra simetrica a cifrei bâlbâite în palindromul creat după bâlbâială. Pentru simplitate, ne vom referi la aceasta ca ”cifra bâlbâită”.(nu este obligatoriu să avem mereu o cifră bâlbâită și o cifră din mijloc, întrucât sunt multe cazuri în care în cel mai mare *palilindrom* care se poate forma nu trebuie sa aibă o singură cifră în mijloc sau o cifră bâlbâită care nu e în mijloc)

- **Soluția 1** - Complexitate timp $O(N * \Sigma^2)$
  Fixam valoarea cifrei care se va bâlbâi și valoarea cifrei care va fi în mijloc.(vom analiza și cazurile în care aceste cifre sunt inexistente).
  Întrucât am fixat, în complexitate $O(\Sigma^2)$, cele 2 cifre, tot ce mai trebuie sa facem este să vedem care este cel mai mare palindrom(de lungime pară) care se poate forma folosind restul cifrelor, lucru care se poate face în complexitate $O(N)$.
  Cifra bâlbâită(care nu e în mijloc) va trebui adăugată în palindromul calculat fie în jumătatea stângă, fie în jumătatea dreaptă. Trebuie analizate ambele cazuri.
  Din toate aceste $palilindroame$ obținute se va afișa maximul.
- **Soluția 2** - Complexitate timp $O(N * \Sigma)$
  Observăm că putem fixa doar valoarea cifrei care se va bâlbâi(și nu e în mijloc) și că nu mai e nevoie să fixăm și valoarea cifrei din mijloc, întrucât aceasta poate fi determinată: este cea mai mare cifră cu frecvența impară!
  Așadar, ne fixăm în complexitate $O(\Sigma)$ valoarea cifrei care se va bâlbâi și apoi vom calcula cel mai mare palindrom(de orice lungime de data aceasta) care se poate forma folosind restul cifrelor, lucru care se poate face în complexitate $O(N)$.
  Din toate aceste $palilindroame$ obținute se va afișa maximul.
- **Soluția 3** - Complexitate timp $O(N)$
  Observăm că nu mai este nevoie sa fixăm valoarea cifrei care se va bâlbâi, întrucât aceasta poate fi determinată: este cea mai mare cifră disponibilă cu frecvența impară și minim 3.
  La această soluție sunt mai multe cazuri de analizat decât la celelalte, întrucât acum trebuie sa determinăm și cifra bâlbâită și cifra din mijloc, iar una dintre determinări o poate influența pe cealaltă.
  Astfel, această soluție are cea mai bună complexitate timp: $O(N)$.

Problema 2: oneout

Propusa de: prof. Cheșcă Ciprian

Soluția 1(brut force) - *prof. Cheșcă Ciprian*
Se determină un nou șir w pentru care $w[i] = 0$ dacă $v[i]$ este liber de pătrate și $w[i] = 1$ dacă $v[i]$ nu este liber de pătrate, folosind descompunerea în factori primi.
Utilizând acest șir se identifică pozițiile elementelor de 1 care au ,,vecini” elemente de 0. Din aceste poziții se extinde secvența către stânga și către dreapta. Se actualizează lungimea maximă dacă este cazul.
Pentru a determina câte secvențe de lungime maximă sunt se parcurge încă odată șirul și apoi încă odată pentru a afișa indicii de start și de final ai secvențelor de lungime maximă. Soluția are ordinul de complexitate $O(n\sqrt{n})$.

Soluția 2 - *prof. Cheșcă Ciprian*
Se determină șirul numerelor libere de pătrate printr-o metoda de tip ciur asemănătoare cu ciurul lui Eratostene.
Se poate folosi algoritmul:
- se inițializează șirul w cu 0 *(presupunem că toate numerele sunt libere de pătrate)*

```
for(i = 1; i <= max; i++)
    w[i] = 0;
for(i = 2; i*i <= max; i++)
    for(j = i*i; j <= max; j += i*i)
        w[j] = 1;
```

(am eliminat toate pătratele perfecte și multiplii acestora, unde max este cea mai mare valoare din șirul v)
Se urmează apoi etapele de la soluția anterioara folosind șirul numerelor libere de pătrate generat anterior.
Deoarece metoda de determinare a șirului numerelor libere de pătrate are ordin de complexitate $O(n \log n)$ soluția obține 100 puncte.

Solutia 3 - *student Gheorghe Liviu Armand*
După ce determinăm dacă un număr e *liber de pătrate*, atunci putem proceda la fel ca la soluțiile descrise anterior pentru a afla numărul de numere *libere de pătrate* și lungimea celei mai lungi *bisecvențe* care respectă condițiile din enunt.

Dar cum aflam dacă un număr x este *liber de pătrate*?
Putem lua în ordine crescătoare fiecare număr prim p și să determinăm dacă x este divizibil cu pătratul lui. Dacă este, atunci x nu este *liber de pătrate* și ne putem opri. Dacă nu este, atunci înseamnă că p se afla fie la puterea 0, fie la puterea 1 în descompunerea lui x în factori primi. Dacă p se află la puterea 1, îl vom împărți pe x la p, întrucât, în acest fel, va mai trebui să parcurgem mai puține numere prime decât dacă nu am face aceasta împărțire.

Principala idee a acestei soluții este următoarea:
Dacă am realiza procedeul acesta pentru toate numerele prime care *ar putea* apărea la puterea a-3-a în descompunerea în factori primi a lui x și dacă toate verificările cu aceste numere prime nu au condus încă la un verdict exact, atunci numărul meu x rămas după eventualele împărțiri poate fi:

- fie un număr prim q la puterea 1 => $x = q$
- fie un număr prim q la puterea 2 => $x = q^2$
- fie un produs de 2 numere prime q si r => $x = q * r$

Observăm că x nu poate să fie un produs de 2 numere prime q și r astfel încât $x = q * r^2$ sau $x = q^2 * r$ sau $x = q^2 * r^2$. Presupunem, prin absurd, că x poate să fie sub una dintre aceste forme. Fără a restrânge generalitatea problemei, presupunem că $q \leq r$. Atunci $q^3 \leq x$. Dar conform procedeului descris mai sus, am eliminat din descompunerea în factori primi a lui x orice număr prim care *ar fi putut* fi la o putere mai mare sau egală cu 3. Așadar, după terminarea procedeului, q nu ar mai fi apărut în descompunerea în factori primi a lui x. Deci, avem o contradicție cu enunțul. Presupunerea făcută este falsă. Așadar, x poate fi numai sub una din cele 3 forme bune enumerate mai sus.

Din cele 3 cazuri bune, observăm că x nu este *liber de pătrate* doar atunci când este un pătrat perfect. Astfel, putem întreba după fiecare pas dacă nu cumva x a devenit pătrat perfect și ne putem opri. Dacă s-a terminat tot procedeul, înseamnă că x este *liber de pătrate*.

O altă optimizare pe care o putem face este ca pentru fiecare număr x întâlnit pe tot parcursul procedeului(deci și cel inițial și cei obținuți după împărțirile care au loc în cadrul procedeului) să ținem minte rezultatul obținut la sfârșitul procedeului(adică facem memoizare). Acest lucru este corect, întrucât observăm că toate numerele x întâlnite pe parcursul unui procedeu sunt de același tip: fie *libere de pătrate*, fie nu sunt *libere de pătrate*.
Astfel, cand vom întâlni pe parcursul procedeului un x deja calculat, vom putea opri procedeul, întrucât știm ce rezultat va da, deoarece ni l-am salvat când l-am întâlnit prima oară.

Așadar, acest procedeu va itera doar prin primele 25 de numere prime(în cazul limitelor din această problemă) în cazul cel mai nefavorabil. În practică se comporta mult mai bine și de multe ori va încheia procedeul înainte să parcurgă cele 25 de numere prime. Această soluție este un pic mai lenta decât soluția anterioară, însa se descurcă foarte bine cu numere mai mari.

În concluzie, această soluție obține 100 de puncte. Are complexitate timp O(N*25), însă se comporta mai bine de atât în practică.

Problema 3: pergament

Propusa de: student Mihaela Cismaru

Soluția 1 - 40 de puncte

Pentru obținerea punctajului se poate implementa o soluție brută în care păstram structura pergamentului într-o matrice de $N \times K$ poziții. Notăm în matrice toate pozițiile prin care trece o stradă verticală sau orizontală. După adăugarea tuturor străzilor, prin parcurgerea matricei poziție cu poziție se pot număra toate intersecțiile dintre cele două tipuri de străzi. Complexitatea acestei soluții este $O(N \cdot K)$.

Soluția 2 - 70 de puncte

Pentru obținerea a încă 30 de puncte este necesară stocarea pe rând în memorie a câte o coloana din pergament, respectiv un vector de N poziții. Pentru fiecare poziție a unei coloane putem verifica dacă prin aceasta trece strada orizontală a rândului de care aparține. Facem transformarea tuturor străzilor verticale ce aparțin de fiecare coloană în poziții ce determină începerea sau terminarea unei străzi verticale. Sortând crescător aceste poziții pentru o coloana și parcurgând de sus în jos, putem știi în timp real dacă ne aflăm pe o stradă verticală sau nu. Adunând toate intersecțiile de pe fiecare coloană obținem astfel numărul final cerut. Complexitatea acestei soluții este tot $O(N \cdot K + Q \log Q)$ dar memoria folosită va fi de doar $O(N + Q)$.

Soluția 3 - 100 de puncte

Pentru obținerea punctajului maxim putem păstra în memorie doar străzile verticale și vectori de dimensiune a doar K poziții. Transformăm datele străzilor verticale în poziții de început și de final a unei străzi verticale și le sortăm după rând (complexitate $Q \log Q$). Parcurgem fiecare rând de sus în jos, ținând minte într-un vector pozițiile în care se afla o stradă verticală și un alt vector în care vom calcula sumele parțiale ale vectorului anterior. La fiecare poziție unde se termină sau începe o stradă verticală vom recalcula cei doi vectori. Numărul acestor evenimente va fi de exact $2Q$ iar recalcularea vectorului de sume parțiale va avea complexitate $O(K)$ în total având o complexitate de doar $O(Q \cdot K)$. Pentru fiecare rând putem afla câte intersecții există interogând intervalul străzi orizontale de pe acel rând în vectorul din sume parțiale, interogare cu o complexitate de doar $O(1)$. Se vor face exact N astfel de interogări, astfel soluția finală va avea complexitatea optimă de $O(N + K \cdot Q + Q \log Q)$

Comisia științifică

- prof. Nodea Gheorghe-Eugen - Colegiul Național ”Tudor Vladimirescu”, Tg. Jiu
- prof. Cheșcă Ciprian - Liceul Tehnologic ”Grigore C. Moisil”, Buzău,
- prof. Candale Silviu - Colegiul Național ”Liviu Rebreanu”, Bistrița Năsăud,
- prof. Ciurea Stelian - Universitatea ”Lucian Blaga”, Sibiu
- student Gheorghe Liviu Armand - Universitatea din București
- student Cismaru Mihaela - NetRom Software, Universitatea din Craiova
- student Iordache Ioan-Bogdan - Universitatea din București
- student Arhire Andrei - Universitatea A.I. Cuza, Iași
- student Popescu Mihai Cristian - Universitatea Babeș-Bolyai Cluj-Napoca
- student Petrescu Alexandru - Universitatea din Oxford