Descrierea Soluțiilor
Concursul Național "InfoPro", Runda 2
Grupa C2

1 Problema cifrevecine

Propunator: prod. Marius Nicoli, C.N. "Frații Buzești" - Craiova

Pentru $k = 1$ soluția se obține eliminând cea mai din stânga cifră după care urmează una mai mare. Dar soluția și pentru acest caz rezultă și din abordarea generală, prezentată mai jos. Numărul de cifre fiind mic, putem încerca toate modalitățile de a elimina k cifre vecine. Eliminăm pe rând ultimele k cifre, apoi secvența anterioară formată din k cifre etc. La un moment dat construim valoarea obținută fără secvența de k cifre curentă și facem efectiv maximul acestor valori. O modalitate de a obține o valoare prin eliminarea unei anume secvență de k cifre dintr-un număr n este descrisă în continuare. Construim în variabila p valoarea 10^k iar în variabila r punem la început valoarea 1. Când trecem de la o secvență de k cifre la alta, înmulțim cu 10 atât p cât și r. Ne interesează valorile: n / p * r + n modulo r.

2 Problema palpow

Propunator: prof. Raluca Costineanu, C.N. "Ștefan cel Mare" - Suceava

Pentru fiecare număr din cele n din șir determinăm: numărul de divizori $nrDiv$, oglinditul numărului ogl, numărul de divizori ai oglinditului $nrDivOgl$, contorizăm numerele pentru care $nrDivOgl > nrDiv$ și tot pentru aceste numere determinăm minimul și maximul. În funcție de modul de determinare a numărului de divizori putem obține punctaje diferite:

I căutăm toți divizorii numărului dorit mai mici decât $nr/2$

```
int nrDiv = 1, d;
for(d = 1; d <= nr / 2; d ++)
    if(nr % d == 0)
    nrDiv ++;
```

II căutăm divizorii numărului dorit „în pereche". (dacă d este divizor pentru nr atunci și nr/d este divizor, și este suficient să căutăm divizorii doar până la $\sqrt{nr}$).

```
int nrDiv = 0, d;
for(d = 1; d * d < nr; d ++)
    if(nr % d == 0)
    nrDiv + = 2;
if(d * d == nr) nrDiv ++;
```

III calculăm numărul de divizori din factorizarea numărului dacă $nr = d1^{p1} * d2^{p2} * ... * dk^{pk}$ atunci $nrDiv = (p1 + 1) * (p2 + 1) * ... * (pk + 1)$

```
int nrDiv = 1, d = 3, p = 0;
while(nr % 2 == 0)
    nr / = 2, p ++;
nrDiv * = p + 1; p = 0;
while(d * d <= nr)
{
    while(nr % d == 0)
        nr / = d, p ++;
    nrDiv * = p + 1;
    d + = 2; p = 0;
}
if(nr > 1)nrDiv * = 2;
```

Pe cazul cel mai defavorabil a II-a și a III-a soluție duc la timp de calcul de ordin $\sqrt{nr}$. Soluția a III-a poate fi îmbunătățită prin precalcularea numerelor prime (de exemplu folosind Ciurul lui Eratostene) și împărțire direct la aceste valori. Această abordare necesită cunoștințe despre tablouri și nu era necesară pentru obținerea punctajului maxim în acest concurs.

3 Problema abk1k2

Propunator: prof. Ionel Vasile Piț Rada, C.N. "Traian"-Drobeta Turnu Severin

3.1 Soluția 1

Cu două repetiții, una în alta, fixăm toate perechile de numere din intervalul $[a, b]$. La fiecare în parte calculăm cel mai mare divizor comun. La o analiză atentă a restricțiilor pentru a și b constatăm că acest algoritm nu se încadrează în timp pentru toate testele.

3.2 Soluția 2

Fie X = numărul de miltipli ai lui $k1$ care se află în intervalul $[a, b]$. X poate fi calculat rapid astfel: $X = b/k1 – (a - 1)/k1$.

Observăm că alegând oricare două numere contorizate la X, formăm o submulțime care ne interesează și care are cel mai mare divizor comun al elementelor sale multiplu de $k1$. Avem $X * (X - 1)/2$ variante de a alege două astfel de numere (suma Gauss, dacă ne gândim că putem cupla primul număr cu toate celelalte $X - 1$, apoi al doilea număr cu celelalte $X - 2$ mai mari ca el etc).

Similar avem $Y * (Y - 1)/2$ submulțimi care au cmmdc multiplu de $k2$.

Trebuie însă să scădem acele submulțimi care au fost numărate și pentru $k1$ și pentru $k2$.

Acestea sunt în număr de $Z * (Z - 1)/2$, unde Z este numărul de valori din intervalul $[a, b]$ care sunt multipli atât ai lui $k1$ cât și ai lui $k2$.

Pentru calculul lui Z putem determina cel mai mic multiplu comun al lui $k1$ și $k2$ (să îl notăm K) și Z este numărul de multipli ai săi din intervalul $[a, b]$ adică $b/K - (a - 1)/K$.

Calculul celui mai mic multiplu comun se face împărțind produsul numerelor la cel mai mare divizor comun al lor. Pentru a ne încadra în timpul de executare impus, la calculul celui mai mare divizor comun trebuie folosit algoritmul lui Euclid.

Echipa care a pregătit setul din probleme pentru această rundă a fost formată din:

- prof. Marius Nicoli, Colegiul Național "Frații Buzești", Craiova
- prof. Raluca Costineanu, Colegiul Național "Ștefan cel Mare", Suceava
- Bogdan Iordache, Universitatea București
- prof. Cristina Iordaiche, Liceul "Grigore Moisil" Timișoara
- prof. Ionel Vasile Piț Rada, Colegiul Național "Traian", Drobeta Turnu Severin