LATVIJAS 24. INFORMĀTIKAS OLIMPIĀDES

I POSMA UZDEVUMU ATRISINĀJUMI

Šaubīga matrica

Cipars X varētu būt attēlots tad un tikai tad, ja visi izslēgtie elementi ciparā X ir izslēgti arī uz tablo. Pēc šī fakta noskaidrošanas atliek tikai uzrakstīt programmu. Taču lai nepārvērstu to par 3 kilometrus garu kodu, un lai nerakstītu kaudzi „if"-u ar lielu iespēju kļūdīties, varētu paspēlēties ar bitu aritmētiku. Sastādīsim tabulu, kur ar 1 apzīmēsim izslēgtos elementus, un ar 0 – ieslēgtos. Pēc tam iegūto bināro skaitli nokonvertēsim.

Cipars	A	B	C	D	E	F	G	Skaitlis
	64	***32***	***16***	***8***	***4***	***2***	***1***	
0	0	0	0	1	0	0	0	**8**
1	1	1	0	1	1	0	1	**109**
2	0	1	0	0	0	1	0	**34**
3	0	1	0	0	1	0	0	**36**
4	1	0	0	0	1	0	1	**69**
5	0	0	1	0	1	0	0	**20**
6	0	0	1	0	0	0	0	**16**
7	0	1	0	1	1	0	1	**45**
8	0	0	0	0	0	0	0	**0**
9	0	0	0	0	1	0	0	**4**

Tagad sakonvertēsim skaitli uz tablo tieši tāda formāta, ka tas ir dots (1 – ieslēgts, 0 – izslēgts). Apzīmēsim šo skaitli ar A. Tagad, ja mēs gribam noskaidrot, vai uz tablo A varētu būt cipars X, tad mēs pielietojam bināro UN (AND) skaitļiem A un X (iegūto tabulā). Ja rezultāts ir 0, tad tas varētu būt cipars X. Ja nav 0 – tad nē. Kāpēc? Atcerēsimies atrisinājumu. Mums ir jāpārliecinās, ka visi izslēgti elementi ciparā X ir izslēgti arī A. Visi izslēgti elementi ciparā X ir apzīmēti ar 1. Visi pārējie ar 0, tātad arī binārā UN rezultātā tajos elementos būs 0. Tātad, vieninieki rezultātā var parādīties tikai elementos, kuri ir izslēgti ciparā X. Tā kā A izslēgtie elementi apzīmēti ar 0, tad, ja visi izslēgtie elementi ciparā X ir izslēgti arī A, vieninieka nekur nebūs, un rezultātā mēs dabūsim 0 – cipars X varētu būt attēlots. Ja nav 0 – tad kāds izslēgts elements ciparā X bija ieslēgts A – cipars X nevar būt attēlots.

Augs

Katrā nedēļā var būt tikai četru veidu zari – papildzars, galvenais zars ar garumu 1, galvenais zars ar garumu 2 un galvenais zars ar garumu 3, kur jau ir uzziedējis zieds. Apzīmēsim šo četru veidu zaru skaitu attiecīgi ar skaitļiem A_0, A_1, A_2 un A_3. Pieņemsim, ka šīs vērtības ir aprēķinātas kādai nedēļai K. Mēģināsim izskaitļot atbilstošos skaitļus B_0, B_1, B_2 un B_3 nākamajai nedēļai K+1.

- A_0 zari sadalās 2 daļas – B_0 un B_1 zaros. Katra jauna tipa zaru ir tik, cik bija A_0 zaru.
- A_1 zari sadalās 2 daļas – B_0 un B_2 zaros. Katra jauna tipa zaru ir tik, cik bija A_1 zaru.
- A_2 zari sadalās 2 daļas – B_0 un B_3 zaros. Katra jauna tipa zaru ir tik, cik bija A_2 zaru.
- A_3 zari nedalās.

Tātad:

- $B_0 = A_0 + A_1 + A_2$
- $B_1 = A_0$
- $B_2 = A_1$
- $B_3 = A_2$

Mums ir zināma situācija 0.-tās nedēļas beigās: $A_0 = 0$, $A_1 = 1$, $A_2 = 0$, $A_3 = 0$.

Tātad, izmantojot iepriekš dotās formulas, varam uzzināt situāciju arī 1., 2., 3., …, N. nedēļas beigās.

Modinātājs

Sākumā ir jānolasa dati. Laika pieraksts sastāv no četrām sastāvdaļām: diena (D = [0 (Pir) .. 6 (Sve)]), stunda (H = [0 .. 23], minūte (M = [0 .. 59]) un sekunde (S = [0 .. 59]). Varam aprēķināt laiku vienā vienībā – sekundēs, kur laiks nozīmēs cik sekundes pagāja no nedēļas sākuma:

- T = D * 24 * 60 * 60 + H * 60 * 60 + M * 60 + S

Tagad atrodam T_a (tagadējo laiku) un T_b (modinātāja laiku). Rezultāts = $T_b - T_a$. Ja rezultāts ir negatīvs, tad tas nozīme, ka modinātājs atskanēs tikai nākamajā nedēļā, un pie rezultāta mums ir jāpieskaita 7 * 24 * 60 * 60. Bet diena var nebūt uzdota (***). Tāda gadījumā nekas daudzi nemainās – mums vienkārši ir septiņi dažādi modinātāja laiki, un no visiem šiem rezultātiem ir jāizvēlas mazākais.

Kačaks

Acīmredzams, ka tā ir aritmētiskā progresija, un mums ir jāatrod minimāls X, lai izpildītos:

- $\frac{X(X+1)}{2} \geq N$

X arī būs atbilde. Protams, šo uzdevumu var atrisināt, atrisinot augšminēto kvadrātnevienādību. Tomēr, eksistē nedaudz lēnāks (bet joprojām pietiekami ātrs) veids, kā šo var atrisināt – binārā meklēšana. Mēs zinām, ka atbilde ir diapazonā $[1..2 * 10^9]$ (to var viegli pārbaudīt). Tātad, mums ir diapazons [L .. R]. Paņemsim skaitli, kas atrodas šī diapazona vidū – C. Ja $\frac{C(C+1)}{2} \geq N$, tad neviens no skaitļiem [C+1 .. R] mūs neinteresē - tie arī apmierina nevienādību, bet C ir mazāks. Ja $\frac{C(C+1)}{2} < N$, tad neviens no skaitļiem [L .. C] mūs neinteresē – neviens no tiem nevar apmierināt nevienādību, jo mazākiem skaitļiem šī izteiksme būs vēl mazāka. Tātad, ar katru gājienu mēs atmetam kā nederīgu pusi no skaitļiem, jeb samazinām diapazonu uz pusēm. Turpinām procesu, kamēr diapazonā paliks viens skaitlis. Tā arī būs atbilde. Viss process atkārtosies ne vairāk ka 31 reizi (kas ir ļoti maz). Risinājuma pseidokods izskatās šādi:

```
L = 1
R = 2 * 10 ^ 9
while R > L:
    C = (L + R) / 2
    if ((C * (C + 1)) / 2) >= N:
          R = C
    if ((C * (C + 1)) / 2) < N:
          L = C + 1
answer = L
```

Parlaments

Vispirms definēsim dažus lielumus:

- A_j^i = cik dažādos veidos no pirmām i partijām var izveidot apvienojumu ar kopējo deputātu skaitu j.

Tagad, zinot A^i, pamēģināsim atrast A^{i+1}. Lai partijas #(i+1) deputātu skaits ir X. Kā mēs varam noskaidrot A_j^{i+1}? Ir divi varianti – ņemt partiju #(i+1) apvienojumā, vai neņemt. Ja neņemam, tad visi j dalībnieki nāk no pirmām i partijām. Ja ņemam, tad X dalībnieki nāk no partijas #(i+1), un j-X dalībnieki nāk no pirmām i partijām. Jāievēro, ka j nevar būt mazāks par X. Tātad:

- $A_j^{i+1} = A_j^i + \begin{cases} A_{j-X}^i, & ja\ j \geq X \\ 0, & ja\ j < X \end{cases}$, kur X – dalībnieku skaits partijā #(i+1)

Zinot, ka:

- $A_j^0 = \begin{cases} 1, & ja\ j = 0 \\ 0, & ja\ j > 0 \end{cases}$,

mēs varam pakāpeniski izrēķināt A^1, A^2, A^3, …, A^N.

- Atbilde $= A^N_{\lfloor \frac{S}{2} \rfloor + 1} + A^N_{\lfloor \frac{S}{2} \rfloor + 2} + A^N_{\lfloor \frac{S}{2} \rfloor + 3} + \cdots + A^N_S$, kur S – kopējais deputātu skaits,

tā kā mūs interesē tikai tie apvienojumi, kuri ir arī koalīcija, jeb tai pieder vairāk nekā puse no kopējo deputātu skaitu.

Jāpiebilst, ka iegūtie skaitļi var būt ļoti lieli, bet mums ir jāizvada tikai atlikums, dalot ar 1000000. Tātad, arī visus skaitļus mēs rēķinām un glabāsim kā atlikumus.

Trijniekiem – nē!

Sākumā iedomāsimies, ka mums nav ierobežojuma uz trijniekiem, un ka **visi** skaitļi tabulā ir pēc kārtas:

y \ x	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**
3	49	26	27	28	29	30	31
2	48	25	10	11	12	13	32
1	47	24	9	2	3	14	33
0	46	23	8	1	4	15	34
-1	45	22	7	6	5	16	35
-2	44	21	20	19	18	17	36
-3	43	42	41	40	39	38	37

Tagad mēs gribam pēc koordinātām atrast kādu noteiktu skaitli. Visa tabula sadalās gredzenos (skat.zīm.). Noskaidrosim, kā iespējams aprēķināt noteiktas rūtiņas gredzena numuru (0. gredzens – sākumrūtiņa).

- Gredzena numurs = $\max(|X|, |Y|)$

Tagad mēs varam noskaidrot gredzena malas garumu, un kāds ir pirmais skaitlis, kas ir uzrakstīts šajā gredzenā:

- Gredzena malas garums = gredzena numurs * 2 + 1
- Pirmais skaitlis gredzenā = $1 + \begin{cases} 0, & ja\ numurs = 0 \\ (malas\ garums - 2)^2, & ja\ numurs > 0 \end{cases}$

Mēs zinām, ka katrā gredzenā skaitļi tiek aizpildīti noteiktā secībā, un tāpēc mēs varam ātri noskaidrot malu, kur atrodas meklētais skaitlis.

Tagad, zinot skaitli X, mums vajag noteikt X.-to skaitli pēc kārtas bez 3 sava pierakstā. Citiem vārdiem, mums vajag noteikt X.-to pēc kārtas skaitli ar 9 iespējamiem cipariem pierakstā. Tātad mums vajag nokonvertēt skaitli X sistēma ar bāzi 9. Pseidokods skaitļu konvertācijai no bāzes 10 uz jebkuru citu bāzi, ir šāds:

```
A = number    //šajā gadījumā X
B = Ø         //rezultāts
base = base   //šajā gadījumā 9
while A > 0:
    B = (A mod base) + B
    A = A / base
```

Bet mums bija jādabū skaitlis bez cipara 3. Šobrīd esam dabūjuši visus skaitļus bez cipara 9 (to nav sistēmā ar bāzi 9). Šī problēma arī ir viegli atrisināma: nepieciešams vienkārši pārsaukt ciparus:

- 0 -> 0, 1 -> 1, 2 -> 2, 3 -> 4, 4 -> 5, 5 -> 6, 6 -> 7, 7 -> 8, 8 -> 9

Tagad ir iegūts prasītais skaitlis.