

Klases spēle

Ievērojama Taškentas vidusskola svin savu jubileju, rīkojot spēli starp skolēniem un skolotājiem. Klasē sēž N skolēni, kas ir sanumurēti no 0 līdz $N - 1$. Katram skolēnam ir papīra lapa, uz kuras ir veselu skaitļu masīvs. Sākotnēji katra lapa ir tukša, tas ir, tajā ir tukšs masīvs.

Skolēni spēlē spēli kopā ar M skolotājiem, kas ir sanumurēti no 0 līdz $M - 1$, un direktoru. Spēle tiek spēlēta M soļos, kas arī ir sanumurēti no 0 līdz $M - 1$. Solī j skolotājs j ienāk klasē, un notiek šādi notikumi:

- Daži (iespējams, nulle) skolēni paceļ roku. Tiek garantēts, ka katrā solī vismaz viens skolēns **nepaceļ** roku, un katrs skolēns visas spēles laikā drīkst pacelt roku **ne vairāk kā vienu reizi**.
- Skolotājs apskata skolēnu pašreizējās lapas un var **modificēt** to skolēnu lapas, kuri šajā solī **nepacēla** roku. Skolotājs var aizstāt katra šāda skolēna lapas saturu ar jaunu (iespējams, tukšu) veselu skaitļu masīvu. Katram vesalam skaitlim jābūt no 0 līdz 63 ieskaitot, un masīvā var būt ne vairāk kā 63 elementi.
- Pēc šīm izmaiņām skolotājs j aiziet, un skolēni apmainās ar savām lapām atbilstoši slepenai permutācijai P . Tas ir, katrs skolēns i ($0 \leq i < N$) nodod savu lapu skolēnam $P[i]$, kur $P[0], P[1], \dots, P[N - 1]$ ir N **atšķirīgi** veseli skaitļi no 0 līdz $N - 1$ (ieskaitot). P vērtības ir **fiksētas** visos spēles posmos, bet skolotāji un direktors tās nezina.

Kad visi M soļi ir pabeigti un pēdējā apmaiņa saskaņā ar P ir veikta, ienāk direktors. Aplūkojot tikai skolēnu turētās lapas, direktoram katram skolēnam i ($0 \leq i < N$) jānosaka soļa numurs j , kurā skolēns i pacēla roku, vai jāsecina, ka skolēns i nekad nav pacēlis roku.

Skolotāji nevar sazināties ar citiem skolotājiem vai direktoru, izņemot caur veselu skaitļu masīviem, kas uzrakstīti uz papīra lapām. Katrs skolotājs zina, kurā solī viņš/viņa ieiet klasē.

Tavs uzdevums ir izstrādāt un ieviest stratēģiju skolotājiem un direktoram, kas pareizi noteiktu, vai un kad katrs skolēns pacēla roku. Tavs rezultāts ir atkarīgs no garākā uz kādas lapas uzrakstītā masīva garuma: ja maksimālais garums ir īsāks, iegūtais punktu skaits ir labāks vai tāds pats.

Implementēšanas detaļas

Tev jāimplementē divas procedūras: viena skolotājiem un viena direktoram.

Procedūra, kas Tev jāimplementē **skolotājiem**, ir šāda:

```
std::vector<std::vector<int>> process_step(
    int N, int M, int R,
    std::vector<int> T,
    std::vector<std::vector<int>> A)
```

- N : skolēnu skaits.
- M : soļu skaits, kas ir arī skolotāju skaits.
- R : pašreizējā soļa numurs (no 0 līdz $M - 1$).
- T : masīvs (iespējams, tukšs), kas satur skolēnu, kas šajā solī paceļ rokas, numurus augošā secībā.
- A : masīvs garumā N ; apraksta lapas, kur $A[i]$ ($0 \leq i < N$) ir veselu skaitļu masīvs, kas rakstīts uz papīra, ko soļa sākumā tur i -tais skolēns.
- Šī procedūra tiek izsaukta tieši M reizes spēlē, secībā $R = 0, 1, \dots, M - 1$.

Procedūrai jāatgriež masīvs B garumā N , kas attēlo papīra lapas pēc skolotāja veiktajām izmaiņām, tādā pašā formātā kā A .

- Katram skolēnam i , kurš pacēla roku (tas ir, i parādās T), papīra lapa nedrīkst tikt mainīta, tātad, $B[i] = A[i]$.
- Katram i , kuram $0 \leq i < N$, $B[i]$ garums nedrīkst pārsniegt 63, un katram veselajam skaitlim iekš $B[i]$ jābūt no 0 līdz 63 (ieskaitot).

Procedūra, kas Tev jāimplementē **direktoram**, ir šāda:

```
std::vector<int> determine_steps(
    int N, int M, std::vector<std::vector<int>> A)
```

- N, M : tie paši, kas iepriekš.
- A : masīvs garumā N , kur $A[i]$ ir veselo skaitļu masīvs, kas rakstīts uz papīra lapas, ko tur i -tais skolēns pēc visiem M soļiem.
- Šī procedūra katrā spēlē tiek izsaukta tieši vienu reizi – pēc pēdējā `process_step` izsaukuma.

Procedūrai ir jāatgriež masīvs D garumā N . Katram i , kuram $0 \leq i < N$, jāizpildās

- $D[i] = j$, ja i -tais skolēns pacēla roku j -tā soļa laikā, vai
- $D[i] = -1$, ja i -tais skolēns visas spēles laikā ne reizi nepacēla roku.

Tavai programmai nav atļauts saglabāt vai pārsūtīt informāciju no viena `process_step` izsaukuma uz citu, izņemot caur `process_step` atgriezto vērtību. Ja Tava programma mēģinās to darīt, Tavs rezultāts CMS var būt nepareizs un var tikt samazināts pēc sacensību beigām. Ņem vērā, ka vērtētājs var vienlaikus izpildīt vairākas Tavas programmas instances. Tas nozīmē, ka `process_step` un `determine_steps` izsaukumi no vienas spēles var tikt izpildīti dažādās

instancēs, un `process_step` un `determine_steps` izsaukumi no dažādām spēlēm var tikt izpildīti vienā instancē. Katrā testpiemērā ir ne vairāk kā 5 spēles.

Ierobežojumi

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Katrs skolēns visas spēles laikā paceļ roku **ne vairāk kā vienu reizi**. Citiem vārdiem, katram skolēnam i ir ne vairāk kā viens solis j , kurā i paceļ roku.
- Katrā solī vismaz viens skolēns **nepaceļ** roku.

Apakšuzdevumi un vērtēšana

Apakšuzdevums	Punkti	Papildu ierobežojumi
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ visiem $0 \leq i \leq N - 1$.
4	25	Katrā solī ne vairāk kā viens skolēns paceļ roku.
5	56	Bez papildu ierobežojumiem.

Katram testa gadījumam Tavs iegūtais punktu skaits ir 0 (CMS sistēmā tiek ziņots kā `Output isn't correct` – izvad dati nav pareizi), ja jebkura procedūras `process_step` izsaukuma atgrieztā vērtība ir nederīga vai ja jebkura procedūras `determine_steps` izsaukuma atgrieztā vērtība ir nepareiza.

Pretējā gadījumā ar C apzīmēsim maksimālo garumu **jebkuram** masīvam **jebkurā** B , ko atgriež `process_step`. Tad testpiemēra punktu skaits apakšuzdevumā, kas vērts S punktus, tiek aprēķināts kā $S \cdot X$, kur X ir atkarīgs no C atbilstoši šai tabulai:

Nosacījums	X
$C \leq 2$	1.00
$C = 3$	0.75
$C = 4$	0.55
$5 \leq C \leq 13$	$0.50 - 0.03 \cdot (C - 5)$
$14 \leq C \leq 63$	$0.19 \cdot \frac{64-C}{64-14} + 0.04$

Piemērs

Aplūkosim gadījumu ar $N = 4$ skolēniem, $M = 2$ skolotājiem, un permutāciju $P = [0, 3, 1, 2]$. Sākumā visas papīra lapas ir tukšas.

Vērtētājs vispirms izsauc:

```
process_step(4, 2, 0, [1], [[], [], [], []])
```

1. skolēns ir pacēlis savu roku, tāpēc viņa papīra lapa nevar tikt mainīta. 0. skolotājs var izlemt uzrakstīt $[10]$ uz 0. skolēna papīra lapas, uzrakstīt $[1, 63, 4]$ uz 3. skolēna papīra lapas, un atstāt 2. lapu tukšu. Lai to izdarītu, procedūrai jāatgriež $B = [[10], [], [], [1, 63, 4]]$.

Pēc tam, kad 0. skolotājs aiziet, skolēni apmaina savas papīra lapas atbilstoši P . Pēc šīs apmaiņas lapas ir $A = [[10], [], [1, 63, 4], []]$.

Vērtētājs tagad izsauc:

```
process_step(4, 2, 1, [0, 3], [[10], [], [1, 63, 4], []])
```

0. un 3. skolēns ir pacēlis roku, tāpēc viņu lapas nevar tikt mainītas. 1. skolotājs var izlemt uzrakstīt $[0, 40]$ uz 1. skolēna lapas un uzrakstīt $[1, 50]$ uz 2. skolēna lapas. Lai to izdarītu, procedūrai jāatgriež $B = [[10], [0, 40], [1, 50], []]$.

Pēc tam, kad 1. skolotājs aiziet, papīra lapas atkal tiek apmainītas atbilstoši P . Pēc apmaiņas lapas ir $A = [[10], [1, 50], [], [0, 40]]$.

Visbeidzot vērtētājs izsauc:

```
determine_steps(4, 2, [[10], [1, 50], [], [0, 40]])
```

Šai procedūrai jāatgriež masīvs $D = [1, 0, -1, 1]$, jo 0. un 3. skolēns pacēla roku 1. solī, 1. skolēns pacēla roku 0. solī, un 2. skolēns nevienā brīdī nepacēla roku. Šajā piemērā $C = 3$.

Paraugvērtētājs

Ievaddati:

```
N M
Q[0] Q[1] ... Q[N-1]
P[0] P[1] ... P[N-1]
```

Šeit Q ir masīvs garumā N , kur $Q[i] = j$, ja i -tais skolēns pacēla roku j -tajā solī, vai $Q[i] = -1$, ja i -tais skolēns nevienā brīdī visas spēles laikā nepacēla roku.

Izvaddatu formāts:

Pēc katra `process_step` izsaukuma paraugvērtētājs izvada uz katras papīra lapas uzrakstīto. Ar K apzīmēsim B garumu un ar $L[i]$ apzīmēsim $B[i]$ garumu (katram $0 \leq i < K$). Paraugvērtētājs izvada B tālāk aprakstītajā formātā, **beigās izvadot tukšu rindu**:

```
K
L[0] B[0][0] B[0][1] ... B[0][L[0]-1]
L[1] B[1][0] B[1][1] ... B[1][L[1]-1]
:
L[K-1] B[K-1][0] B[K-1][1] ... B[K-1][L[K-1]-1]
```

Tad paraugvērtētājs apmaina papīra lapas atbilstoši permutācijai P . Ja $K \neq N$, tad tā vietā paraugvērtētājs izvada kļūdas paziņojumu un beidz darbu.

Pēc `determine_steps` izsaukšanas paraugvērtētājs izvada:

```
C H
D[0] D[1] ... D[H-1]
```

Šeit H ir masīva D garums, ko atgriež `determine_steps`.