

Klasseværelsesspil

Et veletableret gymnasium i Tashkent fejrer deres fødselsdag ved at spille et spil mellem elever og lærere. Der sidder N elever i klasseværelset, nummereret fra 0 til $N - 1$. Hver elev holder et stykke papir hvorpå der er nedskrevet et array af heltal. Til at starte med er alle papirer blanke, altså er der på hvert papir nedskrevet et tomt array.

Eleverne spiller et spil med M lærere, nummereret fra 0 til $M - 1$ og én rektor.

Der er M runder i spillet, som også er nummereret fra 0 til $M - 1$. I den j 'te runde kommer den j 'te lærer ind i klasseværelset og følgende handlinger bliver udført:

- Nogle (muligvis ingen) elever rækker deres hænder op. Det er garanteret at i hver runde at mindst en elev **ikke** rækker hånden op, og hver elev må række hånden op **højst én gang** igennem hele spillet.
- Lærerne kigger på de papirer eleverne på det tidspunkt holder og må **ændre** på papirerne holdt af elever der **ikke** har rækket deres hånd op. For hver elev kan læren erstatte arrayet på papiret med et nyt (muligvis tomt) array af heltal. Hvert heltal skal være et tal mellem 0 og 63, inklusivt, og arrayet kan højst have 63 elementer.
- Efter disse ændringer forlader lærer j klasseværelset og, baseret på en hemmelig permutation P , bytter eleverne deres papirer. Det vil sige at hver elev i ($0 \leq i < N$) giver deres papir til elev $P[i]$, hvor $P[0], P[1], \dots, P[N - 1]$ er N **forskellige** heltal mellem 0 og $N - 1$, inklusivt. Værdierne af P er **uændret** gennem alle runder af spillet, men lærerne og rektor kender dem ikke.

Når alle M runder er færdige og den sidste ombytning baseret på P har fundet sted træder rektor ind i klasseværelset. Ved kun at kigge på papirerne, som eleverne holder, skal rektor, for hver elev i ($0 \leq i < N$), bestemme i hvilken runde elev i rakte hånden op eller konkludere at eleven aldrig rakte hånden op.

Lærerne kan ikke kommunikere med andre lærere eller rektor, undtagen gennem de array af heltal nedskrevet på papirerne. Hver lærer ved hvilken runde det er når de træder ind i klasseværelset.

Din opgave er at finde og implementere en strategi for lærerne og rektor som korrekt identificerer, for hver elev, hvornår de rakte hånden op eller at de ikke rakte hånden op. Dine point afhænger af den maksimale længde af alle papirer som er blevet nedskrevet på et papir, en kortere maximal længde giver flere eller det samme antal point.

Implementations Detaljer

Du skal implementere to funktioner. Én for lærerne og én for rektor.

Funktionen du skal implementere for **lærerne** er:

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

- N : antal elever.
- M : antal skridt og antal lærer.
- R : hvilken runde vi er på (fra 0 til $M - 1$).
- T : et (muligvis tomt) array som består af indekserne af elever som rækker deres hånd op i denne runde, sorteret i stigende rækkefølge.
- A : et array af længde N som beskriver alle parene, hvor $A[i]$ ($0 \leq i < N$) er arrayet af heltal på papiret holdt af den i 'te elev i starten af runden.
- Denne funktion bliver kaldet præcis M gange per spil i rækkefølgen $R = 0, 1, \dots, M - 1$.

Denne funktion skal returnere et array B af længde N , som repræsenterer papirerne, efter lærerens ændringer, i samme format som A .

- For hver elev i der har rækket hånden op (altså i fremgår i T) skal det i 'te papir være uændret, altså skal $B[i] = A[i]$.
- For hver i således at $0 \leq i < N$ skal længden af $B[i]$ ikke overskride 63 og hvert heltal i $B[i]$ skal ligge mellem 0 og 63, inklusivt.

Funktionen som du skal implementere for **rektor** er:

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

- N, M : som ovenover.
- A : et array af længde N , hvor $A[i]$ er arrayet af heltal på papiret holdt af den i 'te elev efter alle M runder.
- Denne funktion bliver kaldt præcis én gang per spil, efter den sidste gang `process_step` bliver kaldt.

Funktionen skal returnere et array D af længde N . For hver i således at $0 \leq i < N$ skal der gælde at

- $D[i] = j$ hvis den i 'te elev rækker deres hånd op i den j 'te runde eller
- $D[i] = -1$ hvis den i 'te elev aldrig rækker hånden op igennem hele spillet.

Dit program må ikke gemme eller videregive noget som helst af information på tværs af kaldene til `process_step`, undtagen gennem den returneret værdi af `process_step`. Hvis dit program forsøger at gøre dette kan dine point i CMS blive reduceret efter konkurrencen er ovre. Læg mærke til at graderen kan køre flere instanser af dit program på samme tid. Det betyder at kald til `process_step` og `determine_steps` fra det samme spil kan køres i forskellige instanser og kald til `process_step` og `determine_steps` fra forskellige spil kan blive kørt i samme instans. Hver test case indeholder op til 5 spil.

Begrænsninger

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Hver elev rækker hånden op **højst én gang** gennem hele spillet. Det betyder at for den i 'te elev er der mindst en runde j i hvilken i rækker hånden op.
- I hver runde er der mindst en elev som **ikke** rækker hånden op.

Delopgaver and Point

Delopgaver	Point	Yderligere begrænsninger
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ for alle $0 \leq i \leq N - 1$.
4	25	Højst en elev rækker deres hånd op i hver runde.
5	56	Ingen yderligere begrænsninger.

For hver test case får du 0 point (tilbagerapporteret som `Output isn't correct` i CMS) hvis den returnerer en ugyldig værdi fra `process_step` eller hvis et kald til `determine_steps` returnerer en forkert værdi.

Ellers, lad C være den maksimale længde af **alle** arrays i **alle** B returneret af `process_step`. Så er pointene for en test case i en delopgave med S point beregnet som $S \cdot X$, hvor X afhænger af C som vist i følgende tabel:

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

Eksempel

Overvej følgende scenarie med $N = 4$ elever, $M = 2$ lærere og permutationen $P = [0, 3, 1, 2]$. Til at starte med er alle papirene blanke.

Graderens første kald:

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

Elev 1 har rækket deres hånd op, så deres papir forbliver som det er. Lærer 0 kan bestemme sig for at skrive $[10]$ på den 0'te elevs papir, skrive $[1, 63, 4]$ på den 3'dje elevs papir og lade 2s papir stå tomt. For at gøre dette skal funktionen returnere $B = [[10], [], [], [1, 63, 4]]$.

Efter lærer 0 går ud af klasseværelset bytter eleverne deres papirer ved at følge P . Efter ombytningen af papirene er $A = [[10], [], [1, 63, 4], []]$.

Graderen kalder nu:

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

Elev 0 og 3 har rækket hånden op, så deres papirer forbliver som de er. Lærer 1 må vælge at skrive $[0, 40]$ på elev 1's papir og $[1, 50]$ på elev 2's papir. For at gøre dette skal funktionen returnere $B = [[10], [0, 40], [1, 50], []]$.

Til slut kalder graderen

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

Denne funktion skal returnere arrayet $D = [1, 0, -1, 1]$ da eleverne 0 og 3 har rækket hånden op i runde 1, elev 1 har rækket hånden op i runde 0 og elev 2 aldrig har rækket hånden op. I dette eksempel er $C = 3$.

Eksempel Grader

Input format:

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

Her er Q et array af længde N , hvor $Q[i] = j$ hvis elev i rækker hånden op i løbet af runde j eller $Q[i] = -1$ hvis den i 'te elev aldrig rækker hånden op i løbet af spillet.

Output format:

Efter hvert kald til `process_step`, outputter eksempel graderen indholdet af papirerne. Lad K være længden af B og $L[i]$ være længden af $B[i]$ (for hvert $0 \leq i < K$). Eksempel graderen printer B i følgende format, **efterfulgt af en tom linje**:

```
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]
```

Eksempe graderen bytter så papirerne ved at følge P . Hvis $K \neq N$ printer graderen en fejlbesked og terminere, i stedet, programmet.

Efter at have kaldet `determine_steps` printer eksempel graderen:

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

Her er H længden af arrayet D , returneret af `determine_steps`.