

Celotřídní hra

Central Asian University oslavuje své výročí pořádáním zápasu mezi studenty a učiteli. Ve třídě sedí N studentů očíslovaných od 0 do $N - 1$. Každý student drží list papíru, který obsahuje pole celých čísel. Zpočátku je každý list prázdný, takže obsahuje prázdné pole.

Studenti hrají hru s M učiteli očíslovanými od 0 do $M - 1$, a také s ředitelem školy. Hra se hraje v M krocích, které jsou také číslovány od 0 do $M - 1$. V kroku j vstoupí učitel j do třídy a dojde k následujícím událostem:

- Někteří (možná žádní) studenti zvednou ruku. Je zaručeno, že v každém kroku alespoň jeden student **nezvedne** ruku, a každý student může zvednout ruku **maximálně jednou** za celou hru.
- Učitel se podívá na papíry, které studenti aktuálně drží, a může **upravit** papíry, které drží studenti, kteří v tomto kroku **nezvedli** ruku. Pro každého takového studenta může učitel nahradit obsah jeho papíru novým (možná prázdným) polem celých čísel. Každé celé číslo musí být v rozmezí od 0 do 63 včetně a pole může mít maximálně 63 prvků.
- Po těchto úpravách učitel j odejde a studenti si vymění své papíry podle tajné permutace P . To znamená, že každý student i ($0 \leq i < N$) předá svůj papír studentovi $P[i]$, kde $P[0], P[1], \dots, P[N - 1]$ je N **různých** celých čísel v rozmezí 0 až $N - 1$ včetně. Hodnoty P jsou během všech kroků hry **pevně dané**, ale učitelé ani ředitel je neznají.

Jakmile je dokončeno všech M kroků a byla provedena poslední výměna dle permutace P , vstoupí do místnosti ředitel školy. Ředitel vidí pouze papíry, které drží studenti, a musí u každého studenta i ($0 \leq i < N$) určit číslo kroku j , ve kterém student i zvedl ruku, nebo dojít k závěru, že student i ruku nikdy nezvedl.

Učitelé nemohu komunikovat s ostatními učiteli ani s ředitelem jinak než prostřednictvím posloupností celých čísel zapsaných na papírech. Každý učitel zná číslo kroku, ve kterém vstupuje do třídy.

Vaším úkolem je navrhnout a implementovat strategii pro učitele a ředitele, která správně určí, zda a kdy každý student zvedl ruku. Vaše bodování bude záviset na maximální délce všech polí zapsaných na papíře: kratší maximální délka znamená vyšší nebo stejné množství bodů.

Implementační detaily

Potřebujete implementovat dvě funkce, jednu pro učitele a druhou pro ředitele.

Funkce, kterou byste měli implementovat pro **učitele**, je:

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

- N : počet studentů.
- M : počet kroků a zároveň počet učitelů.
- R : číslo aktuálního kroku (od 0 do $M - 1$).
- T : (možná prázdné) pole sestávající z indexy studentů, kteří v tomto kroku zvedli ruku, seřazené vzestupně.
- A : pole délky N popisující papíry, kde $A[i]$ ($0 \leq i < N$) je pole celých čísel na papíru, který drží student i na začátku kroku.
- Tato funkce je volána právě M -krát za hru, v pořadí $R = 0, 1, \dots, M - 1$.

Funkce by měla vrátit pole B délky N , představující papíry po úpravách učitele, ve stejném formátu jako A .

- Pro každého studenta i , který zvedl ruku (tj. i je prvkem T), nesmí být papír změněn, tedy musí platit $B[i] = A[i]$.
- Pro každé i , pro které platí $0 \leq i < N$, nesmí délka $B[i]$ překročit 63 a každé celé číslo v $B[i]$ musí ležet v rozmezí 0 až 63 včetně.

Funkce, kterou byste měli implementovat pro **ředitele školy**, je následující:

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

- N, M : totéž jako výše.
- A : pole o délce N , kde $A[i]$ je pole celých čísel na papíře, který drží student i po všech M krocích.
- Tato funkce se volá přesně jednou za hru, a to po posledním volání funkce `process_step`.

Funkce musí vrátit pole D o délce N . Pro každé i (pro které $0 \leq i < N$) musí platit, že:

- $D[i] = j$ pokud student i zvedl ruku během kroku j , nebo
- $D[i] = -1$ pokud student i nezvedl ruku ani jednou za celou hru.

Váš program nesmí ukládat ani přenášet žádné informace mezi jednotlivými voláními funkce `process_step`, s výjimkou informace předané skrz návratovou hodnotu funkce `process_step`. Pokud se o to váš program pokusí, může být vaše bodování v systému CMS nesprávné a po skončení soutěže může být sníženo. Vezměte na vědomí, že grader může spustit více instancí vašeho programu současně. To znamená, že volání na `process_step` a `determine_steps` ze stejné hry mohou být provedeny v různých instancích, zatímco volání na `process_step` a `determine_steps` z různých her mohou být provedeny ve stejné instanci. Každý test může obsahovat až 5 her.

Omezení

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Každý student zvedne ruku **nanejvýš jednou** za celou hru. Jinými slovy, pro každého studenta i existuje nanejvýš jeden krok j , ve kterém i zvedne ruku.
- V každém kroku alespoň jeden student ruku **nezvedne**.

Podúlohy a bodování

Podúloha	Body	Omezení
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ pro každé $0 \leq i \leq N - 1$.
4	25	V každém kroku zvedne ruku nanejvýš jeden student.
5	56	Žádná další omezení.

U každého testu je váš počet bodů 0 (v CMS hlášeno jako `Output isn't correct`), pokud je návratová hodnota jakéhokoli volání funkce `process_step` neplatná nebo pokud je návratová hodnota jakéhokoli volání funkce `determine_steps` nesprávná.

V opačném případě necht' C je maximální délka **všech** polí ve **všech** polích B vrácených funkcí `process_step`. Body pro testovací případ v podúloze s možným počtem bodů S se pak vypočítá jako $S \cdot X$, kde X závisí na C podle následující tabulky:

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

Příklad

Uvažujme scénář s $N = 4$ studenty, $M = 2$ učiteli a permutací $P = [0, 3, 1, 2]$. Zpočátku jsou všechny papíry prázdné.

Grader nejprve zavolá:

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

Student 1 zvedl ruku, takže jeho papír již nelze upravovat. Učitel 0 se může rozhodnout, že na papír studenta 0 napíše $[10]$, na papír studenta 3 napíše $[1, 63, 4]$ a papír studenta 2 nechá prázdný. K tomu musí funkce vrátit $B = [[10], [], [], [1, 63, 4]]$.

Poté, co učitel 0 odejde, si studenti vymění své papíry podle P . Po výměně jsou papíry $A = [[10], [], [1, 63, 4], []]$.

Grader nyní zavolá:

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

Studenti 0 a 3 zvedli ruku, takže jejich papíry nelze měnit. Učitel 1 se může rozhodnout, že na papír studenta 1 napíše $[0, 40]$ a na papír studenta 2 napíše $[1, 50]$. K tomu musí funkce vrátit hodnotu $B = [[10], [0, 40], [1, 50], []]$.

Poté, co učitel 1 odejde, dojde k opětovné výměně papírů podle permutace P . Po výměně jsou papíry v tomto pořadí: $A = [[10], [1, 50], [], [0, 40]]$.

Na závěr testovací grader zavolá:

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

Tato funkce musí vrátit pole $D = [1, 0, -1, 1]$, protože studenti 0 a 3 zvedli ruku v kroku 1, student 1 zvedl ruku v kroku 0 a student 2 vůbec nezvedl ruku. V tomto příkladu platí, že $C = 3$.

Ukázkový grader

Formát vstupu:

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

Zde Q je pole délky N , kde $Q[i] = j$ pokud student i zvedne ruku během kroku j , nebo $Q[i] = -1$, pokud student i nezvedne ruku ani jednou za celou hru.

Formát výstupu:

Po každém volání funkce `process_step` vzorový grader vypíše obsah papírů. Necht' K je délka pole B a $L[i]$ je délka pole $B[i]$ (pro každé $0 \leq i < K$). Ukázkový grader vypíše B v následujícím formátu, **následovaném prázdným řádkem**:

```
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]
```

Ukázkový grader poté vymění papíry podle permutace P . Pokud platí $K \neq N$, ukázkový grader místo toho vypíše chybovou zprávu a skončí.

Po volání funkce `determine_steps` ukázkový grader vypíše:

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

Zde H je délka pole D , které vrací funkce `determine_steps`.