Класстык оюн

Ташкенттеги белгилүү орто мектеп өзүнүн маарекесин окуучулар менен мугалимдердин ортосунда оюн өткөрүп, белгилеп жатат. Класста 0 дөн $N-1$ге чейин номерленген N окуучу отурушат. Ар бир окуучунун колунда бүтүн сандар массиви камтылган кагаз бар. Башында ар бир кагаз бош, ошондуктан анын ичинде бош массив бар.

Окуучулар оюнду 0 дөн $M-1$ ге чейин номерленген M жана мектептин директору менен ойноп жатышат. Оюн M кадам менен ойнолот жана 0 дөн $M-1$ ге чейин номерленет. Оюндун j - кадамында, j - мугалим класска кирет жана төмөнкү окуялар болот:

- Айрым (балким нөл) окуучу колун көтөрөт. Ар бир кадамда жок дегенде бир окуучу колун **көтөрбөйт** жана ар бир окуучу оюн учурунда колун **эң көп дегенде бир жолу** көтөрө алат.
- Мугалим окуучулардын учурда кармап турган кагаздарын карап чыгат жана бул кадамда кол көтөрбөгөн окуучулардын кагаздарын **өзгөртүшү** мүмкүн. Мындай ар бир окуучу үчүн мугалим кагазынын мазмунун жаңы (мүмкүн бош) бүтүн сандар массиви менен алмаштыра алат. Ар бир бүтүн сан 0 жана 63 ортосунда болушу керек, ал эми массивде эң көп дегенде 63 элемент болушу мүмкүн.
- Бул өзгөртүүлөрдөн кийин, j-мугалим кетип, окуучулар P жашыруун пермутациясына ылайык өз кагаздарын алмашышат. Башкача айтканда, ар бир i ($0 \leq i < N$) окуучусу өзүнүн кагазын $P[i]$ окуучусуна берет, мында N сан $P[0], P[1], \ldots, P[N-1]$ жана сандар 0 менен $N-1$дин ортосундагы **ар кандай** бүтүн сандар. P маанилери оюндун бардык кадамында **бирдей** болот, бирок мугалимдер жана мектеп директору аларды билишпейт.

Бардык M кадам бүткөндөн жана P боюнча акыркы алмашуу жүргүзүлгөндөн кийин, директор кирет. Окуучулардын колундагы кагаздарды гана карап чыгып, директор ар бир i ($0 \leq i < N$) окуучу үчүн i- окуучусу колун көтөргөн j- кадамынын санын аныкташы керек же i - окуучусу эч качан колун көтөрбөгөн деген тыянакка келиши керек.

Мугалимдер башка мугалимдер же мектеп директору менен баарлаша алышпайт, бирок барактарга жазылган бүтүн сандар массивдери аркылуу гана байланышса болот. Ар бир мугалим класска кайсы кадамдан киргенин билет.

Сиздин милдетиңиз - мугалимдер жана мектеп директору үчүн ар бир окуучу колун качан жана качан көтөргөнүн туура аныктоочу стратегияны иштеп чыгуу жана ишке ашыруу. Сиздин упайыңыз кагазга жазылган ар кандай массивдин максималдуу узундугуна жараша болот: кыска максималдуу узундук жогорку же барабар упай берет.

Ишке ашыруунун чоо-жайы

Сиз эки жол-жобону ишке ашырышыңыз керек, бири мугалимдер үчүн, экинчиси директор үчүн.

Мугалимдер үчүн сиз колдонушуңуз керек болгон жол-жобо:

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

- N : окуучулардын саны.
- M : кадамдардын саны жана мугалимдердин саны.
- R : учурдагы кадамдын номери (0 дөн $M-1$ге чейин).
- T : бул кадамда кол көтөргөн окуучулардын индекстеринен турган, өсүү тартибинде иреттелген (бош болушу мүмкүн) массив.
- A : кагаздарды сүрөттөгөн N узундуктагы массив, мында $A[i]$ ($0 \leq i < N$) - кадамдын башында i окуучусу кармап турган кагаздагы бүтүн сандардын массиви.
- Бул процедура ар бир оюнда M жолу деп аталат, $R = 0, 1, \ldots, M-1$ тартибинде.

Процедура мугалимдин өзгөртүүлөрүнөн кийинки макалаларды A форматында көрсөткөн N$ узундуктагы B массивин кайтарышы керек.

- Колун көтөргөн ар бир i окуучу үчүн (башкача айтканда, i T дан чыгат) кагазы өзгөртүлбөшү керек, ошондуктан $B[i] = A[i]$.
- $0 \leq i < N$ болгон ар бир i үчүн, $B[i]$ узундугу 63 ашпашы керек жана $B[i]$ ичиндеги ар бир бүтүн сан, анын ичинде 0 жана 63 ортосунда болушу керек.

Директор үчүн сиз ишке ашырышыңыз керек болгон жол-жобо:

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

- N , M : жогорудагыдай эле.
- A : узундугу N болгон массив, мында $A[i]$ M кадамдын баарынан кийин i окуучуну кармап турган кагаздагы бүтүн сандардын массиви.
- Бул процедура ар бир оюнда бир жолу, `process_step` акыркы жолу чакыргандан кийин чакырылат.

Процедура узундугу N болгон D массивин кайтарышы керек. $0 \leq i < N$ болгон ар бир i үчүн, ал төмөнкүлөрдү аткарышы керек

- $D[i] = j$ эгер i окуучусу j кадамында колун көтөрсө, же
- Эгер i окуучусу оюн учурунда колун көтөрбөсө $D[i] = -1$.

Программаңыз бир чакыруудан `process_step` функциясына экинчисине эч кандай маалыматты сактоого же берүүгө уруксат берилбейт, `process_step` кайтарым мааниси аркылуу гана. Эгерде программаңыз муну жасоого аракет кылса, CMS'теги упайыңыз туура эмес болушу мүмкүн жана сынак аяктагандан кийин төмөндөшү мүмкүн. Баалоочу программаңыздын бир нече инстанцияларын бир эле учурда аткарышы мүмкүн экенин эске алыңыз. Бул бир эле оюндан `process_step` жана `determine_steps` функцияларына чакыруулар ар кандай инстанцияларда аткарылышы мүмкүн экенин, ал эми ар кандай оюндардан `process_step` жана `determine_steps` функцияларына чакыруулар бир эле учурда аткарылышы мүмкүн экенин билдирет. Ар бир тест учуру 5 чейинки оюндарды камтыйт.

Чектөөлөр

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Ар бир окуучу оюн учурунда колун **эң көп дегенде бир жолу** көтөрөт. Башкача айтканда, ар бир i окуучусу үчүн, i колун көтөргөн эң көп дегенде бир j кадам бар.
- Ар бир кадамда жок дегенде бир окуучу колун **көтөрбөйт**.

Кошумча тапшырмалар жана упайлар

Кошумча тапшырма	Баалоо	Кошумча чектөөлөр
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ бардык $0 \leq i \leq N - 1$ үчүн.
4	25	Ар бир кадамда эң көп дегенде бир окуучу колун көтөрөт.
5	56	Кошумча чектөөлөр жок.

Ар бир тест учуру үчүн, эгерде кандайдыр бир `process_step` процедурасына чакыруунун кайтарым мааниси жараксыз болсо же кандайдыр бир `determine_steps` процедурасына чакыруунун кайтарым мааниси туура эмес болсо, сиздин упайыңыз 0 болот (CMSте `Output isn't correct` деп көрсөтүлөт).

Болбосо, C бул **any** B ичиндеги `process_step` аркылуу кайтарылган **any** массивинин максималдуу узундугу болсун. Андан кийин, S упайы бар кошумча тапшырмадагы тест учуру үчүн упайы $S \cdot X$ катары эсептелет, мында X төмөнкү таблицага ылайык C дан көз каранды:

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

Мисал

$N = 4$ окуучусу,$M = 2$ мугалими жана permutation $P = [0, 3, 1, 2]$ бар сценарийди карап көрүңүз. Башында бардык кагаздар бош.

Баалоочу биринчи чакырат:

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

1 - окуучу кол көтөргөн, ошондуктан анын кагазын өзгөртүүгө болбойт. 0 - мугалим 0 - окуучунун кагазына $[10]$, 3 - окуучу кагазына $[1, 63, 4]$ жазууну жана 2 - окуучунун кагазыны бош калтырууну чечиши мүмкүн. Муну жасоо үчүн процедура $B = [[10], [], [], [1, 63, 4]]$ деп кайтарышы керек.

0 - vугалим кеткенден кийин, окуучулар P боюнча кагаздарын алмаштырышат. Алышып бүткөндөн кийин кагаздар$A = [[10], [], [1, 63, 4], []]$ болуп калат.

Баалоочу эми чакырат:

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

0 жана 3 - окуучулар кол көтөрүштү, ошондуктан алардын кагаздары өзгөртүлбөйт. 1 - мугалим 1- окуучусунун кагазына $[0, 40]$ деп, ал эми 2 - окуучусунун кагазына $[1, 50]$ деп жазууну чечиши мүмкүн. Муну жасоо үчүн процедура $B = [[10], [0, 40], [1, 50], []]$ массивин кайтарышы керек.

1 - мугалим кеткенден кийин, кагаздар кайрадан P ыкмасына ылайык алмаштырылат. Алышып бүткөндөн кийин кагаздар $A = [[10], [1, 50], [], [0, 40]]$ болуп калат.

Акырында баалоочу чакырат:

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

Бул процедура $D = [1, 0, -1, 1]$ массивин кайтарышы керек, анткени 0 жана 3 - окуучулар 1 - кадамда кол көтөрүшкөн,1- окуучу 0 - кадамда кол көтөргөн, ал эми 2 - окуучу эч качан кол көтөргөн эмес. Бул мисалда $C = 3$.

Үлгү грейдер

Киргизүү форматы:

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

Бул жерде Q — узундугу N болгон массив, анда $Q[i] = j$ — эгерде i - окуучу j - кадам учурунда колун көтөрсө, же $Q[i] = -1$ — эгер i - окуучу бүт оюн бою колун көтөрбөсө.

Чыгаруу форматы:

`process_step` функциясына ар бир чакыруудан кийин, үлгү баалоочу документтердин мазмунун чыгарат. K B узундугу жана $L[i]$ $B[i]$ узундугу болсун (ар бир $0 \leq i < K$ үчүн). Үлгү баалоочу B функциясын төмөнкү форматта басып чыгарат, **андан кийин бош сап** жазылат:

```
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]
```

Андан кийин үлгү баалоочу документтерди P пермутациясына ылайык алмаштырат. Эгерде $K \neq N$ болсо, үлгү баалоочу ката билдирүүсүн басып чыгарып, анын ордуна аяктайт.

`determine_steps` чакыргандан кийин, үлгү баалоочу төмөнкүнү чыгарат:

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

Бул жерде, H - `determine_steps` аркылуу кайтарылган D массивинин узундугу.