

Magic City

Ташкент қаласының әкімі әйгілі Magic City ойын-сауық саябағын қайта жобалауды қалайды. Сізге оң бүтін сан K берілген және сіздің міндетіңіз – келесі әдістермен саябақты жобалау.

- Аттракциондар саны N таңдаңыз және аттракциондарды 0 -ден $N - 1$ -ке дейінгі бүтін сандармен белгілеңіз.
- Әртүрлі аттракциондар жұптары арасында екі жақты жолдарды қосыңыз. Бірдей аттракциондар жұбы арасында бірнеше жол болуға рұқсат етіледі. Жолдар арқылы барлық аттракциондар жұптары арасында жүріп өту мүмкіндігі болуы шарт емес.
- $0 \leq i < N$ болатын әрбір i үшін i аттракционға $T[i]$ түрлері тағайындалсын. 0 -ден $2K - 1$ -ге дейін нөмірленген $2K$ түрлері қолжетімді. Әрбір түрі кемінде бір аттракционға тағайындалуы тиіс. Әртүрлі аттракциондарда бірдей түрі болуы мүмкін.

Нарықты зерттеу кезінде мыналар анықтады:

1. Әрбір келуші үш аттракционға барғысы келеді, бірақ бірдей түрдегі екі аттракционға қатарынан бармайды.
2. Келушілер көптеген жолмен қосылған аттракциондарды ұнатпайды.

Әкім барлық келуі мүмкін қонақтарды қанағаттандырғысы үшін сіздің жобаңызға екі шарт қояды.

Біз $0 \leq t_1, t_2, t_3 < 2K$ үшін (t_1, t_2, t_3) деп аталатын түрлердің ретті үштігін **қызықты** деп есептейміз, егер $t_1 \neq t_2$ және $t_2 \neq t_3$. t_1 мен t_3 тең болуы мүмкін екенін ескеріңіз. Осындай $2K \cdot (2K - 1)^2$ қызықты үштік бар.

1-шарт: Әрбір қызықты үштік (t_1, t_2, t_3) үшін келесі шарттарды қанағаттандыратын a_1, a_2, a_3 ($0 \leq a_1, a_2, a_3 < N$ -пен) дейтін үш аттракцион болуы тиіс:

- a_1, a_2, a_3 -ның түрлері t_1, t_2, t_3 -ға сәйкес келуі тиіс. Яғни, $T[a_1] = t_1$, $T[a_2] = t_2$ және $T[a_3] = t_3$.
- a_1 мен a_2 арасында тік жол болуы тиіс.
- a_2 мен a_3 арасында тік жол болуы тиіс.

a_1 пен a_3 арасында жол бар ма, жоқ па, маңызды емес. Сондай-ақ, $t_1 = t_3$ кезінде a_1 мен a_3 аттракциондары бірдей болуы мүмкін екенін ескеріңіз.

2-шарт: Әрбір аттракцион **көп дегенде** K жолдың шеткі нүктесі бола алады.

Бұл тапсырма **partial scoring** жүйесімен бағаланатын 50 тек шығаруға арналған (**output-only**) ішкі есептен тұрады. Әрбір ішкі есеп K -нің нақты бір мәніне сәйкес келеді, және сіз сол K мәні үшін жоғарыда аталған барлық шарттарды қанағаттандыратын саябақты жобалауыңыз керек. Сіздің ұпайыңыз шешіміңіздегі аттракциондар санына байланысты – аттракциондар саны аз болған сайын ұпай жоғары болады.

Implementation Details

Шешіміңізді тапсырудың екі әдісі бар, және әрбір ішкі есеп үшін олардың біреуін пайдалана аласыз:

- **Функция шақыруы**
- **Файл**

Шешіміңізді **функция шақыруы** арқылы жіберу үшін келесі функцияны жүзеге асыруыңыз керек:

```
std::pair<std::vector<int>, std::vector<std::pair<int, int>>>  
construct(int K)
```

- K : аттракцион түрлерінің санының жартысы және әрбір аттракцион шеткі нүктесі болатын жолдарының ең көп саны.
- Бұл функция әрбір ішкі есеп үшін дәл бір рет шақырылады.

Функция ойын-сауық саябағын сипаттайтын (T, E) жұбын қайтаруы тиіс. M – саябағыңыздағы жолдар саны.

- T – ұзындығы N болатын, аттракцион түрлерін сипаттайтын массив.
- E – ұзындығы M болатын, жолдарды сипаттайтын массив. Әрбір $0 \leq j < M$ үшін $E[j] = (U[j], V[j])$ – әртүрлі аттракциондар $U[j]$ мен $V[j]$ арасындағы екі бағытты жол.

Шешіміңізді шығыс файл арқылы тапсыру үшін келесі форматта мәтіндік файл жасап, тапсырыңыз:

```
N M  
T[0] T[1] ... T[N-1]  
U[0] V[0]  
U[1] V[1]  
...  
U[M-1] V[M-1]
```

Ескерту: Сіздің шешіміңіз **дұрыс** деп есептелуі үшін келесі шектеулерді орындауы тиіс:

- $N \leq 2000$

- Әрбір $0 \leq i < N$ үшін $0 \leq T[i] < 2K$, және әрбір аттракцион түрі кемінде бір аттракционға тағайындалуы тиіс.
- Әрбір $0 \leq j < M$ үшін $0 \leq U[j], V[j] < N$ және $U[j] \neq V[j]$.
- 1-ші және 2-ші шарттар орындалуы тиіс.

Constraints

- $1 \leq K \leq 50$

Scoring

Есепте 50 ішкі есеп бар. 1-ден 50-ге дейін әрбір K бір ішкі есепке сәйкес келеді. i -ші ішкі есеп ($1 \leq i \leq 50$) үшін K мәні i -ға тең.

Әрбір ішкі есептің ұпайы S және керекті аттракциондар саны P төмендегі кестеге сәйкес берілген.

Ішкі есептер	S	P
1	1	2
2	8	12
3	9	24
4	9	40
5	9	50
6 - 10	4	$12 \cdot K$
11 - 12	3	$12 \cdot K$
13 - 50	1	$12 \cdot K$

Әрбір ішкі есеп үшін егер сіздің шешіміңіз жарамды ойын-сауық саябағын сипаттамаса, онда сіздің шешіміңіздің ұпайы 0 болады (CMS жүйесінде `Output isn't correct` деп көрсетіледі).

Әйтпесе, сіздің ұпайыңыз N және S мен P параметрлері негізінде келесідей есептеледі:

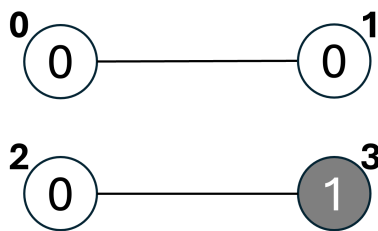
Шарт	Ұпай
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Example

Төмендегі шақыруларды қарастырайық:

```
construct(1)
```

Бұл мысалда $K = 1$, сондықтан $2K = 2$ түрдегі аттракциондар бар. Төмендегі сурет $N = 4$ аттракцион мен $M = 2$ жолдары бар дұрыс шешімді көрсетеді. 0, 1, 2 аттракциондар 0 түріне жатады, ал 3-ші аттракцион 1 түріне жатады.



Екі қызықты үштік бар:

- $(0, 1, 0)$ типі үшін үштік ретінде $(a_1, a_2, a_3) = (2, 3, 2)$ таңдауға болады.
- $(1, 0, 1)$ типі үшін үштік ретінде $(a_1, a_2, a_3) = (3, 2, 3)$ таңдауға болады.

Бұл 1-шарт орындалатынын білдіреді.

Функция $[[0, 0, 0, 1], [(0, 1), (2, 3)]]$ жұбын қайтара алады. Айта кетейік, осы мысалдағы берілген шешім $K = 1$ үшін оңтайлы болмауы мүмкін.

Sample Grader

Input format:

```
K
```

Output format:

```
N M
T[0] T[1] ... T[N-1]
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]
```

Үлгіні грейлер шығысы шығыс файл үшін қажетті форматқа сәйкес келетінін ескеріңіз.