Сыйкырдуу шаар

Ташкенттин мэри атактуу "Сыйкырдуу шаар" аттракцион паркын кайрадан долбоорлоону каалайт. Сизге K оң бүтүн сан берилген жана сиздин тапшырмаңыз - паркты төмөнкүдөй долбоорлоо:

- Аттракциондордун санын камтыган бүтүн сан болгон N тандаңыз. Аттракциондор 0 дөн $N-1$ге чейинки бүтүн сандар менен белгиленген.
- Эки багыттуу жөө жүргүнчүлөр өтүүчү жолдорду кошуңуз, ар бири эки **өзгөчө** аттракциондун ортосуна. Бир эле аттракциондун ортосунда бирден ашык жөө жүргүнчүлөр өтүүчү жолдун болушуна уруксат берилет. **Жөө жүргүнчүлөр өтүүчү жолдор аркылуу ар бир эки аттракциондун ортосунда саякаттоо мүмкүнчүлүгү талап кылынбайт.**
- $0 \le i < N$ болгондой ар бир i үчүн, $T[i]$ тибиндеги i тартуу күчүн дайындаңыз. 0 дөн $2K-1$ге чейин номерленген $2K$ түрлөрү бар. Ар бир түр жок дегенде бир аттракционго таандык болушу керек. Ар кандай аттракциондордун түрү бирдей болушу мүмкүн.

Рыноктук изилдөө төмөнкүлөрдү көрсөттү:

1. Ар бир конок бир эле типтеги эки аттракционго катары менен барбай, үч аттракционго барат.
2. Коноктор көптөгөн жөө жүргүнчүлөр өтүүчү жолдордун акыркы чекити болгон аттракциондорду жактырышпайт.

Бардык потенциалдуу коноктордү канааттандыруу үчүн, мэр сиздин дизайныңызга эки шарт коёт.

Биз $0 \le t_1, t_2, t_3 < 2K$ үчүн **түрлөрдүн** иреттелген үчтүктөр (t_1, t_2, t_3) деп атайбыз. эгерде $t_1 \neq t_2$ жана $t_2 \neq t_3$ болсо, **кызыктуу**. t_1 жана t_3 барабар болушу мүмкүн экенин эске алыңыз. Ошентип, $2K \cdot (2K-1)^2$ кызыктуу үчтүк сандар бар.

1-шарт:

Ар бир кызыктуу үчтүктөр (t_1, t_2, t_3) үчүн, төмөнкүдөй үч аттракцион болушу керек: a_1, a_2, a_3 ($0 \le a_1, a_2, a_3 < N$ менен):

- a_1, a_2, a_3 түрлөрү t_1, t_2, t_3 менен дал келет. Башкача айтканда, $T[a_1] = t_1$, $T[a_2] = t_2$ жана $T[a_3] = t_3$.
- a_1 жана a_2 ортосунда жөө жүргүнчүлөр жолу бар.

- a_2 менен a_3 ортосунда жөө жүргүнчүлөр жолу бар.

a_1 жана a_3 ортосунда жөө жүргүнчүлөр жолу бар же жок экени маанилүү эмес. Ошондой эле $t_1 = t_3$ болгондо, a_1 жана a_3 аттракциондору бирдей болушу мүмкүн экенин эске алыңыз.

2-шарт:

Ар бир аттракцион **эң көп дегенде** K жөө жүргүнчүлөр өтүүчү жолдордун акыркы чекити болушу мүмкүн.

Бул тапшырма **жарым-жартылай бааланган** 50 **чыгарылыш гана** кошумча тапшырмалардан турат. Ар бир кошумча тапшырма K белгилүү бир маанисине туура келет жана сиз K мааниси үчүн жогорудагы бардык шарттарды канааттандырган паркты иштеп чыгышыңыз керек. Сиздин упайыңыз чечимиңиздеги аттракциондордун санына жараша болот : азыраак аттракциондор жогорку же барабар упай берет.

Ишке ашыруунун чоо-жайы

Чечимиңизди тапшыруунун эки ыкмасы бар жана ар бир кошумча тапшырма үчүн кайсынысын болбосун колдонсоңуз болот:

- **Процедура боюнча чакыруу**
- **Чыгаруу файлы**

Чечимиңизди **процедура чакыруу** аркылуу жөнөтүү үчүн, сиз төмөнкү процедураны аткарышыңыз керек:

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

- K : аттракцион түрлөрүнүн санынын жарымы, ошондой эле ар бир аттракциондун акыркы чекити боло ала турган жөө жүргүнчүлөр өтүүчү жолдордун максималдуу саны.
- Бул процедура ар бир кошумча тапшырма үчүн бир гана жолу чакырылат.

Процедура оюн-зоок паркын сүрөттөгөн (T, E) жубун кайтарышы керек. Паркыңыздагы жөө жүргүнчүлөр өтүүчү жолдордун саны M болсун.

- T : аттракциондордун түрлөрүн сүрөттөгөн N узундуктагы массив.
- E : жөө жүргүнчүлөр өтүүчү жолдорду сүрөттөгөн M узундуктагы массив. Ар бир $0 \leq j < M$ үчүн, $E[j] = (U[j], V[j])$ ар башка аттракциондордун ортосундагы $U[j]$ жана $V[j]$ ортосундагы эки багыттуу жөө жүргүнчүлөр өтүүчү жолду билдирет.

Чечимиңизди **чыгаруучу файл** аркылуу жөнөтүү үчүн, төмөнкү форматта тексттик файл түзүп, жөнөтүңүз:

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

Чечимиңиз **жарамдуу** деп эсептелиши үчүн төмөнкү чектөөлөргө жооп бериши керектигин эске алыңыз:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ ар бир $0 \leq i < N$ үчүн, жана ар бир түрү жок дегенде бир аттракционго дайындалышы керек.
- $0 \leq U[j], V[j] < N$ жана $U[j] \neq V[j]$ ар бир $0 \leq j < M$ үчүн.
- 1 жана 2-шарттар аткарылышы керек.

Чектөөлөр

- $1 \leq K \leq 50$

Упай топтоо

50 кошумча тапшырмалары бар, 1 дан 50 чейинки ар бир K бүтүн саны үчүн бирден. i ($1 \leq i \leq 50$) кошумча тапшырмасы үчүн, K функциясынын мааниси i болуп саналат.

Төмөндөгү таблицага ылайык, ар бир кошумча тапшырманын S упайы жана максаттуу тартуулардын саны P болот.

Кошумча тапшырмалар	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	$12 \cdot K$

Ар бир кошумча тапшырма үчүн, эгерде сиздин чечимиңиз жарактуу оюн-зоок паркын сүрөттөбөсө, анда сиздин чечимиңиздин упайы 0 болот (CMSте `Output isn't correct` деп көрсөтүлөт).

Болбосо, сиздин упайыңыз N жана S жана P параметрлеринин негизинде төмөнкүдөй эсептелет:

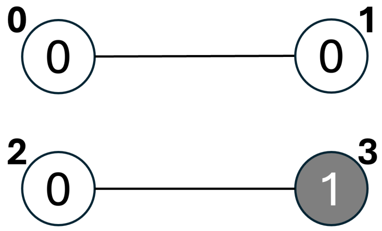
Шарт	Баалоо
$N \le P$	S
$P < N \le 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \le 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Мисал

Төмөнкү чакырууну карап көрүңүз:

```
construct(1)
```

Бул мисалда, $K = 1$, демек, $2K = 2$ аттракциондордун түрлөрү бар. Төмөнкү сүрөттө $N = 4$ аттракциондор жана $M = 2$ жөө жүргүнчүлөр жолдору менен жарактуу чечим көрсөтүлгөн. $0, 1, 2$ аттракциондор 0 түрүндө, ал эми 3 аттракциондор 1 түрүндө болот.



Эки кызыктуу үчтүк бар:

- Үчтүктөр $(0, 1, 0)$ түрү үчүн биз $(a_1, a_2, a_3) = (2, 3, 2)$ тандай алабыз.
- Үчтүктөр $(1, 0, 1)$ түрү үчүн биз $(a_1, a_2, a_3) = (3, 2, 3)$ тандай алабыз.

Бул 1-шарт канааттандырылганын билдирет.

Процедура $([0, 0, 0, 1], [(0, 1), (2, 3)])$ жубун кайтарып бере алат. Бул мисал үчүн берилген чечим $K = 1$ үчүн оптималдуу болбошу мүмкүн экенин эске алыңыз.

Үлгү грейдер

Киргизүү форматы:

```
K
```

Чыгаруу форматы:

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

Үлгү грейдердин чыгышы чыгаруу файлы үчүн талап кылынган форматка дал келерин эске алыңыз.