

Magic City

Taškentas mērs vēlas pārplānot slaveno atrakciju parku "Magic City". Tev ir dots naturāls skaitlis K , un Tavs uzdevums ir izveidot parka plānu šādi:

- Izvēlies veselu skaitli N – atrakciju skaitu. Atrakcijas apzīmē ar veseliem skaitļiem no 0 līdz $N - 1$.
- Pievieno divvirzienu takas; katra savieno divas **atšķirīgas** atrakcijas. Vienu un to pašu atrakciju pāri drīkst savienot vairāk nekā viena taka. **Spēt ceļot starp jebkuru atrakciju pāri, izmantojot takas, nav obligāti.**
- Katram i , kuram $0 \leq i < N$, piešķir atrakcijai i veidu $T[i]$. Ir $2K$ veidi, kas sanumurēti no 0 līdz $2K - 1$. Katram veidam jābūt piešķirtam vismaz vienai atrakcijai. Dažādām atrakcijām var būt viens un tas pats veids.

No tirgus izpētes tika secināts, ka:

- Katrs apmeklētājs vēlas apmeklēt trīs atrakcijas, neapmeklējot divas viena veida atrakcijas pēc kārtas.
- Apmeklētājiem nepatīk atrakcijas, kas ir daudzu gājēju taku galapunkti.

Lai apmierinātu **visus** potenciālos apmeklētājus, mērs pieprasa, lai Tavam plānam izpildās divi nosacījumi.

Sakārtotu **veidu** trijnieku (t_1, t_2, t_3) visiem $0 \leq t_1, t_2, t_3 < 2K$ saucsim par **interesantu**, ja $t_1 \neq t_2$ un $t_2 \neq t_3$. Ņem vērā, ka t_1 var būt vienāds ar t_3 . Tātad, ir $2K \cdot (2K - 1)^2$ interesanti trijnieki.

1. nosacījums: Katram interesantam trijniekam (t_1, t_2, t_3) jābūt trim atrakcijām a_1, a_2, a_3 (kur $0 \leq a_1, a_2, a_3 < N$), tādām, ka:

- a_1, a_2, a_3 veidi atbilst t_1, t_2, t_3 . Tas ir, $T[a_1] = t_1$, $T[a_2] = t_2$, un $T[a_3] = t_3$.
- Starp a_1 un a_2 ir taka.
- Starp a_2 un a_3 ir taka.

Nav svarīgi, vai starp a_1 un a_3 ir taka. Ņem vērā arī to, ka, ja $t_1 = t_3$, atrakcijas a_1 un a_3 var būt vienādas.

2. nosacījums: Katra atrakcija var būt galapunkts **ne vairāk kā** K takām.

Šis uzdevums sastāv no 50 **atvērto ievaddatu** apakšuzdevumiem, kur var iegūt **daļēju punktu skaitu**. Katrs apakšuzdevums atbilst noteiktai K vērtībai, un Tev ir jāizplāno parks, kas atbilst

visiem iepriekš minētajiem nosacījumiem šai K vērtībai. Tavs rezultāts ir atkarīgs no atrakciju skaita Tavā risinājumā: mazāks atrakciju skaits dod augstāku vai tādu pašu rezultātu.

Implementēšanas detaļas

Savu risinājumu var iesniegt divos veidos, un katram apakšuzdevumam vari izmantot jebkuru no tiem:

- **Procedūras izsaukums**
- **Izvaddatu fails**

Lai iesniegtu savu risinājumu, izmantojot **procedūras izsaukumu**, Tev jāimplementē šāda procedūra:

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

- K : puse no atrakciju veidu skaita, kā arī katrai atrakcijai maksimālais gājēju taku, kuru galapunkts tā var būt, skaits.
- Šī procedūra tiek izsaukta tieši vienu reizi katram apakšuzdevumam.

Procedūrai jāatgriež pāri (T, E) , kas apraksta atrakciju parku. Ar M apzīmēsim taku skaitu Tavā parkā.

- T : masīvs garumā N , kas apraksta atrakciju veidus.
- E : masīvs garumā M , kas apraksta takas. Katram $0 \leq j < M$, $E[j] = (U[j], V[j])$ apzīmē divvirzienu taku starp atšķirīgām atrakcijām $U[j]$ un $V[j]$.

Lai iesniegtu savu risinājumu, izmantojot **izvaddatu failu**, izveido un iesniedz teksta failu šādā formātā:

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

Nem vērā, ka, lai Tavs risinājums tiktu uzskatīts par **derīgu**, tam jāizpilda šādi nosacījumi:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ katram $0 \leq i < N$, un katram veidam jābūt piešķirtam vismaz vienai atrakcijai.
- $0 \leq U[j], V[j] < N$ un $U[j] \neq V[j]$ katram $0 \leq j < M$.
- Jāizpildās 1. un 2. nosacījumam.

Ierobežojumi

- $1 \leq K \leq 50$

Vērtēšana

Ir 50 apakšuzdevumi, viens katram veselajam skaitlim K no 1 līdz 50. i -tajam apakšuzdevumam ($1 \leq i \leq 50$), K vērtība ir i .

Iegūstamo punktu skaits S un vēlamais atrakciju skaits P katram apakšuzdevumam ir parādīts tabulā tālāk.

Apakšuzdevumi	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	$12 \cdot K$

Katram apakšuzdevumam, ja Tavs risinājums neapraksta derīgu atrakciju parku, tad Tava risinājuma vērtējums būs 0 (CMS sistēmā tas tiks ziņots kā `Output isn't correct` – izvad dati nav pareizi).

Pretējā gadījumā Tavs punktu skaits tiek aprēķināts, ņemot vērā N un parametrus S un P šādi:

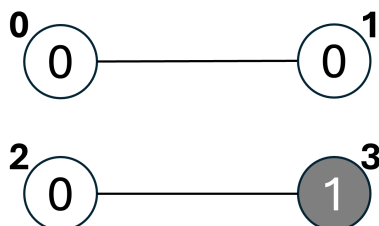
Nosacījums	Punkti
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Piemērs

Aplūkosim šādu izsaukumu:

construct (1)

Šajā piemērā $K = 1$, tāpēc ir $2K = 2$ atrakciju veidi. Tālāk redzamajā attēlā ir parādīts derīgs risinājums ar $N = 4$ atrakcijām un $M = 2$ gājēju takām. 0., 1. un 2. atrakcija ir 0. veida, un 3. atrakcija ir 1. veida.



Ir divi interesanti trijnieki:

- Veidu trijniekam $(0, 1, 0)$ varam izvēlēties $(a_1, a_2, a_3) = (2, 3, 2)$.
- Veidu trijniekam $(1, 0, 1)$ varam izvēlēties $(a_1, a_2, a_3) = (3, 2, 3)$.

Tas nozīmē, ka 1. nosacījums ir apmierināts.

Procedūra var atgriezt pāri $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Ņem vērā, ka šim piemēram dotais risinājums var nebūt optimāls, ja $K = 1$.

Paraugvērtētājs

Ievaddatu formāts:

K

Izvaddatu formāts:

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

Ņem vērā, ka paraugvērtētāja izvaddati atbilst nepieciešamajam izvaddatu faila formātam.