

Magic City

De burgemeester van Tasjkent wil het beroemde Magic City pretpark vernieuwen. Je krijgt een positief geheel getal K en jouw taak is om een park te ontwerpen als volgt:

- Kies een geheel getal N , het aantal attracties. De attracties zijn gelabeld met gehele getallen van 0 tot en met $N - 1$.
- Voeg (ongerichte) wandelpaden toe, steeds tussen twee **verschillende** attracties. Het is toegestaan om meer dan één wandelpad te hebben tussen twee attracties. **Het is niet noodzakelijk dat je tussen elke twee attracties kan reizen langs de wandelpaden.**
- Voor elke i zodat $0 \leq i < N$, geef attractie i een type $T[i]$. Er zijn $2K$ types, genummerd van 0 tot en met $2K - 1$. Er moet van elk type minstens één attractie zijn. Verschillende attracties kunnen hetzelfde type hebben.

Marktonderzoek heeft twee dingen laten zien:

1. Elke bezoeker wil drie attracties bezoeken, zonder twee attracties van hetzelfde type na elkaar te bezoeken.
2. Bezoekers houden niet van attracties die een eindpunt zijn van veel wandelpaden.

Om **alle** mogelijke bezoekers blij te maken stelt de burgemeester twee eisen aan jouw ontwerp.

We noemen een geordend drietal van **types** (t_1, t_2, t_3) met $0 \leq t_1, t_2, t_3 < 2K$ **interessant** als $t_1 \neq t_2$ en $t_2 \neq t_3$. Merk op dat t_1 mogelijk gelijk is aan t_3 . Er zijn dus $2K \cdot (2K - 1)^2$ interessante dretallen.

Eis 1: Voor elk interessant drietal (t_1, t_2, t_3) moeten er drie attracties a_1, a_2, a_3 zijn (met $0 \leq a_1, a_2, a_3 < N$) zodat:

- De types van a_1, a_2, a_3 zijn t_1, t_2, t_3 . Dus $T[a_1] = t_1$, $T[a_2] = t_2$, en $T[a_3] = t_3$.
- Er is een wandelpad tussen a_1 en a_2 .
- Er is een wandelpad tussen a_2 en a_3 .

Het maakt niet uit of er een wandelpad is tussen a_1 en a_3 . Merk ook op dat als $t_1 = t_3$, de attracties a_1 en a_3 mogelijk gelijk zijn.

Eis 2: Elke attractie kan een eindpunt zijn van **maximaal** K wandelpaden.

Deze taak bestaat uit 50 **output-only** subtaken met **deelscores**. Elke subtaak correspondeert met een specifieke waarde van K , en je moet een park ontwerpen dat voldoet aan alle bovenstaande

voorwaarden voor die waarde van K . De score hangt af van het aantal attracties in jouw oplossing: minder attracties geven een hogere of gelijke score.

Implementatiedetails

Er zijn twee manieren om jouw oplossing in te leveren, en je kan voor elke subtaak kiezen welke je gebruikt:

- **Functie-aanroep**
- **Uitvoerbestand**

Om de oplossing in te leveren met een **functie-aanroep**, moet je de volgende functie implementeren:

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

- K : de helft van het aantal types attracties, en ook het maximale aantal wandelpaden waarvan elke attractie het eindpunt kan zijn.
- Deze functie wordt precies één keer aangeroepen voor elk subtaak.

De functie moet een paar (T, E) teruggeven dat een pretpark beschrijft. Noem M het aantal wandelpaden in het park.

- T : een lijst van lengte N die de types van de attracties bevat.
- E : een lijst van lengte M die de wandelpaden geeft. Voor elke $0 \leq j < M$ representeert $E[j] = (U[j], V[j])$ een (ongericht) wandelpad tussen verschillende attracties $U[j]$ en $V[j]$.

Om en oplossing in te leveren met een **outputbestand** moet je een tekstbestand maken en inleveren van de volgende vorm:

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

Merk op dat een oplossing aan de volgende voorwaarden moet voldoen om **geldig** te zijn:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ voor elke $0 \leq i < N$, en van elk type moet er minstens één attractie zijn.
- $0 \leq U[j], V[j] < N$ en $U[j] \neq V[j]$ voor elke $0 \leq j < M$.
- Aan eisen 1 en 2 moet voldaan worden.

Beperkingen

- $1 \leq K \leq 50$

Scores

Er zijn 50 subtaken, een voor elk geheel getal K van 1 tot en met 50. Voor subtaak i ($1 \leq i \leq 50$) is K gelijk aan i .

Elke subtaak heeft een score S en een streefwaarde voor het aantal attracties P , te zien in de tabel hieronder.

Subtaken	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	$12 \cdot K$

Voor elke subtaak geldt dat als de oplossing geen valide pretpark beschrijft, dat de score dan 0 is (weergegeven als `Output isn't correct` in CMS).

Anders wordt je score berekend op basis van N en de parameters S and P als volgt:

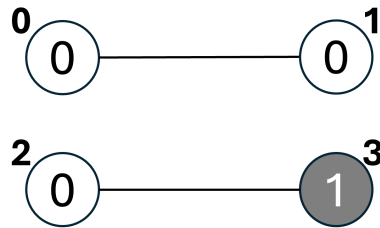
Voorwaarde	Score
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Voorbeeld

Bekijk de volgende aanroep:

```
construct (1)
```

In dit voorbeeld geldt $K = 1$, dus zijn er $2K = 2$ types attracties. De afbeelding hieronder laat een geldige oplossing zien met $N = 4$ attracties en $M = 2$ wandelpaden. Attracties 0, 1, 2 zijn van type 0, en attractie 3 is van type 1.



Er zijn drie interessante drietallen:

- Voor het drietal types $(0, 1, 0)$ kunnen we $(a_1, a_2, a_3) = (2, 3, 2)$ kiezen.
- Voor het drietal types $(1, 0, 1)$ kunnen we $(a_1, a_2, a_3) = (3, 2, 3)$ kiezen.

Dit betekent dat aan eis 1 voldaan wordt.

De functie kan het paar $([0, 0, 0, 1], [(0, 1), (2, 3)])$ teruggeven. Merk op dat de gegeven oplossing voor dit voorbeeld mogelijk niet optimaal is voor $K = 1$.

Voorbeeldgrader

Invoerformaat:

K

Uitvoerformaat:

```

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

Merk op dat de uitvoer van de voorbeeldgrader voldoet aan het formaat van het uitvoerbestand.