

Magisk Stad

Göteborgs president (Quarl Anka) vill bygga om Liseberg. Du får ett positivt heltal K , och ditt uppdrag är att bygga ett nöjesfält på följande vis:

- Välj ett heltal N som anger antalet attraktioner. Attraktionerna är märkta med heltal från 0 till $N - 1$.
- Lägg till dubbelriktade gångvägar, var och en mellan ett par av **olika** attraktioner. Det är tillåtet att ha mer än en gångväg mellan samma par attraktioner. **Det krävs inte att man ska kunna ta sig mellan varje par av attraktioner med hjälp av gångvägarna.**
- För varje i så att $0 \leq i < N$, tilldela attraktion i en typ $T[i]$. Det finns $2K$ typer numrerade från 0 till $2K - 1$. Varje typ ska tilldelas minst en attraktion. Olika attraktioner kan ha samma typ.

Från marknadsundersökningar vet vi att:

1. Varje besökare föredrar att besöka tre attraktioner utan att besöka två attraktioner av samma typ i följd.
2. Besökarna ogillar attraktioner som ligger vid slutet av många gångvägar.

För att göra **alla** potentiella besökare glada ställer Göteborgs president två krav på ditt nöjesfält.

Vi kallar en ordnad trippel av **typer** (t_1, t_2, t_3) för $0 \leq t_1, t_2, t_3 < 2K$ **intressant** om $t_1 \neq t_2$ och $t_2 \neq t_3$. Observera att t_1 kan vara lika med t_3 . Det finns alltså $2K \cdot (2K - 1)^2$ intressanta tripplar.

Krav 1: För varje intressant trippel (t_1, t_2, t_3) måste det finnas tre attraktioner a_1, a_2, a_3 (med $0 \leq a_1, a_2, a_3 < N$) så att:

- Typerna för a_1, a_2, a_3 stämmer överens med t_1, t_2, t_3 . Det vill säga, $T[a_1] = t_1$, $T[a_2] = t_2$ och $T[a_3] = t_3$.
- Det finns en gångväg mellan a_1 och a_2 .
- Det finns en gångväg mellan a_2 och a_3 .

Det spelar ingen roll om det finns en gångväg mellan a_1 och a_3 eller inte. Observera också att när $t_1 = t_3$, kan attraktionerna a_1 och a_3 vara samma.

Krav 2: Varje attraktion kan vara en ändpunkt av **som mest** K gångvägar.

Denna uppgift består av 50 stycken **output-only**-deluppgifter med **delpoäng**. Varje deluppgift motsvarar ett specifikt värde på K , och du måste bygga en park som uppfyller alla ovanstående

villkor för just det värdet på K . Din poäng beror på antalet attraktioner i din lösning: färre attraktioner ger högre eller lika hög poäng.

Implementationsdetaljer

Det finns två sätt att skicka in din lösning, och då får använda valfri för vardera deluppgift:

- Funktionsanrop
- Utdata-fil

För att skicka in din lösning genom ett **funktionsanrop**, så ska du implementera följande funktion:

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

- K : hälften av antalet attraktionstyper, och också det största antalet gångvägar som varje attraktion kan vara en ändpunkt av.
- Denna funktion anropas exakt en gång för varje deluppgift.

Funktion ska returnera ett par (T, E) som beskriver ett nöjesfält. Låt M vara antalet gångvägar i nöjesfältet.

- T : en lista av längd N som beskriver typen av varje attraktion.
- E : en lista av längd M som beskriver gångvägarna. För varje $0 \leq j < M$, $E[j] = (U[j], V[j])$ representerar en dubbelriktad gångväg mellan olika attraktioner $U[j]$ och $V[j]$.

För att skicka in din lösning genom en **udata-fil**, skapa och skicka in en textfil i följande format:

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

Notera att din lösning måste tillfredsställa följande krav för att vara **giltig**:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ för varje $0 \leq i < N$, och varje typ ska tilldelas till minst en attraktion.
- $0 \leq U[j], V[j] < N$ and $U[j] \neq V[j]$ för varje $0 \leq j < M$.
- Krav 1 och 2 måste tillfredställas.

Begränsningar

- $1 \leq K \leq 50$

Poängsättning

Det finns 50 deluppgifter, en för varje heltal K från 1 till 50. För deluppgift i ($1 \leq i \leq 50$) är värdet på K i .

Varje deluppgift har en poäng S och ett måltal attraktioner P enligt tabellen nedan.

Deluppgifter	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	$12 \cdot K$

För varje deluppgift gäller att om din lösning inte beskriver ett giltigt nöjesfält kommer poängen för din lösning att vara 0 (rapporteras som Output isn't correct i CMS).

I övriga fall beräknas din poäng utifrån N och parametrarna S och P enligt följande tabell:

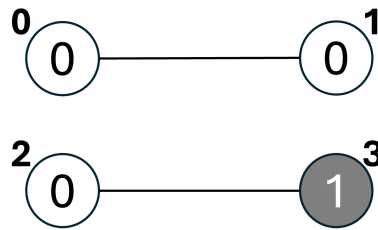
Krav	Poäng
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Exempel

Betrakta följande funktionsskrik:

```
construct(1)
```

I det här exemplet är $K = 1$, så det finns $2K = 2$ typer av attraktioner. Bilden nedan visar en giltig lösning med $N = 4$ attraktioner och $M = 2$ gångvägar. Attraktioner 0,1,2 har typ 0, och attraktion 3 har typ 1.



Det finns två intressanta trippletter:

- För typtrippetten $(0, 1, 0)$ kan vi välja $(a_1, a_2, a_3) = (2, 3, 2)$.
- För typtrippetten $(1, 0, 1)$ kan vi välja $(a_1, a_2, a_3) = (3, 2, 3)$.

Detta innebär att villkor 1 är uppfyllt.

Funktionen kan returnera paret $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Observera att den angivna lösningen i detta exemplet är inte nödvändigtvis optimal för $K = 1$.

Exempeldomare

Inputformat:

K

Outputformat:

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

Notera att outputen från exempeldomaren matchar det angivna formatet för utdatafiler.