

Magic City

Ordføreren i Tashkent ønsker å redesigne den kjente fornøylesparken Magic City. Du blir gitt et positivt heltall K og din oppgave er å designe en fornøylespark som følger:

- Velg et heltall N , antallet attraksjoner. Attraksjonene er nummerert med heltall 0 til $N - 1$.
- Legg til bidireksjonelle gangstier, hver mellom et par av **distinkte** attraksjoner. Det er lov å ha mer en én gangsti mellom samme par med attraksjoner. **Det er ikke påkrevd at man kan gå mellom alle par av attraksjoner ved hjelp av gangstiene.**
- For hver i slik at $0 \leq i < N$, gi attraksjon i en type $T[i]$. Det finnes $2K$ typer nummerert fra 0 til $2K - 1$. Hver type må tildeles minst én attraksjon. Ulike attraksjoner kan ha samme type.

Markedsundersøkelser har vist at:

1. Hver besøkende ønsker å besøke tre attraksjoner uten å besøke to attraksjoner av samme type etter hverandre.
2. Besøkende misliker attraksjoner som er endepunktet til mange gangstier.

For å tilfredsstille **alle** potensielle besøkende har ordføreren innført to begrensninger på designet ditt.

Vi kaller en ordnet trippel av **typer** (t_1, t_2, t_3) der $0 \leq t_1, t_2, t_3 < 2K$ **interessant** hvis $t_1 \neq t_2$ og $t_2 \neq t_3$. Merk at t_1 kan være lik t_3 . Dermed er det $2K \cdot (2K - 1)^2$ interessante tripler.

Begrensning 1: For hver interessante trippel (t_1, t_2, t_3) må det finnes tre attraksjoner a_1, a_2, a_3 (med $0 \leq a_1, a_2, a_3 < N$) sånn at:

- Typene av a_1, a_2, a_3 er lik t_1, t_2, t_3 . Altså, $T[a_1] = t_1$, $T[a_2] = t_2$, og $T[a_3] = t_3$.
- Det finnes en gangsti mellom a_1 og a_2 .
- Det finnes en gangsti mellom a_2 og a_3 .

Det er irrelevant hvorvidt det finnes en gangsti a_1 og a_3 . Merk også at dersom $t_1 = t_3$, så kan attraksjonene a_1 og a_3 være den samme.

Begrensning 2: Hver attraksjon kan være endepunktet til **maksimalt** K gangstier.

Denne oppgaven består av 50 **output-only**-deloppgaver med **partiell poenggivning**. Hver deloppgave korresponderer til en spesifikk verdi av K , og du må da designe en park som møter begrensningene over for denne verdien av K . Poengsummen din avhenger av antallet attraksjoner i løsningen: færre attraksjoner gir en høyere eller lik poengsum.

Implementasjonsdetaljer

Det er to metoder for å levere løsningen din, du kan velge for hver deloppgave.

- **Funksjonskall**
- **Output-fil**

For å levere med **funksjonskall** må du implementere følgende funksjon:

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

- K : halvparten av antallet attraksjonstyper, også det maksimale antallet gangstier en attraksjon kan være endepunkt til.
- Denne funksjonen kalles én gang per deloppgave.

Funksjonen skal returnere et par (T, E) som beskriver en fornøylespark. La M være antallet gangstier i parken din.

- T : en liste av lengde N som beskriver typene til attraksjonene.
- E : en liste av lengde M som beskriver gangstiene. For hver $0 \leq j < M$, representerer $E[j] = (U[j], V[j])$ en bidireksjonell gangsti mellom de distinkte attraksjonene $U[j]$ og $V[j]$.

For å levere ved hjelp av en **output-fil** kan du lage og levere en tekstfil med følgende format:

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

Merk at løsningen din må oppfylle følgende krav for å være **gyldig**:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ for hver $0 \leq i < N$. Det må finnes minst én attraksjon av hver type.
- $0 \leq U[j], V[j] < N$ og $U[j] \neq V[j]$ for hver $0 \leq j < M$.
- Begrensningene 1 og 2 må være oppfylt

Begrensninger

- $1 \leq K \leq 50$

Poeng

Det er 50 deloppgaver, en for hvert heltall K fra 1 til 50. For deloppgave i ($1 \leq i \leq 50$) er verdien av K lik i .

Hver deloppgave har en poengsum S og et ønsket antall attraksjoner P gitt i tabellen under.

Deloppgave	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	$12 \cdot K$

For hver deloppgave, hvis løsningen din ikke beskriver en gyldig fornøylespark, vil poengsummen bli 0 (rapportert som `Output isn't correct` i CMS).

Ellers blir poengsummen beregnet basert på N og parametrene S og P på følgende måte:

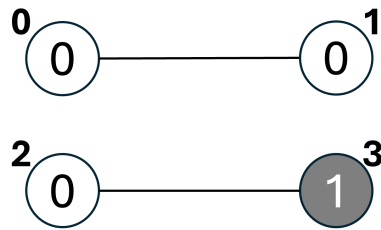
Betingelse	Poeng
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Eksempel

Gitt følgende kall:

```
construct(1)
```

I dette eksempelet er $K = 1$, så det er $2K = 2$ typer attraksjoner. Figuren under viser en gyldig løsning med $N = 4$ attraksjoner $M = 2$ gangstier. Attraksjonene 0, 1, 2 er av type 0, og attraksjon 3 av type 1.



Det er to interessante tripler:

- For trippelen $(0, 1, 0)$ kan vi velge $(a_1, a_2, a_3) = (2, 3, 2)$.
- For trippelen $(1, 0, 1)$ kan vi velge $(a_1, a_2, a_3) = (3, 2, 3)$.

Dette betyr at begrensning 1 er oppfylt.

Funksjonen kan returnere paret $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Merk at eksempelløsningen ikke nødvendigvis er den optimale for $K = 1$.

Sample Grader

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]
```

Merk at outputen til sample graderen er lik formatet til output-fila.