

Cidade Mágica

O prefeito de Tashkent quer reformular o famoso parque de diversões Cidade Mágica. Dado um número inteiro positivo K , sua tarefa é projetar um parque da seguinte forma:

- Escolha um número inteiro N , o número de atrações. As atrações são identificadas por números inteiros de 0 a $N - 1$.
- Adicionar passarelas bidirecionais, cada uma ligando um par de atrações **distintas**. É permitido haver mais de uma passarela entre o mesmo par de atrações. **Não é obrigatório poder se deslocar entre todos os pares de atrações utilizando as passarelas.**
- Para cada i tal que $0 \leq i < N$, atribua à atração i um tipo $T[i]$. Existem $2K$ tipos numerados de 0 a $2K - 1$. Cada tipo deve ser atribuído a pelo menos uma atração. Atrações diferentes podem ter o mesmo tipo.

A pesquisa de mercado revelou que:

1. Cada visitante prefere visitar três atrações, sem visitar duas atrações do mesmo tipo consecutivamente.
2. Os visitantes não gostam de atrações que são conectadas por muitas passarelas.

O prefeito quer satisfazer **todos** os potenciais visitantes e impõe duas condições ao seu projeto.

Chamamos de tripla ordenada de **tipos** (t_1, t_2, t_3) para $0 \leq t_1, t_2, t_3 < 2K$ **Interessante** se $t_1 \neq t_2$ e $t_2 \neq t_3$. Observe que t_1 pode ser igual a t_3 . Portanto, existem $2K \cdot (2K - 1)^2$ triplas interessantes.

Condição 1: Para cada tripla interessante (t_1, t_2, t_3) , devem existir três atrações a_1, a_2, a_3 (com $0 \leq a_1, a_2, a_3 < N$) tais que:

- Os tipos de a_1, a_2, a_3 correspondem a t_1, t_2, t_3 . Ou seja, $T[a_1] = t_1$, $T[a_2] = t_2$ e $T[a_3] = t_3$.
- Existe uma passarela entre a_1 e a_2 .
- Existe uma passarela entre a_2 e a_3 .

É irrelevante se existe ou não uma passarela entre a_1 e a_3 . Observe também que, quando $t_1 = t_3$, as atrações a_1 e a_3 podem ser as mesmas.

Condição 2: Cada atração pode ser o ponto final de **no máximo** K passarelas.

Esta tarefa consiste em 50 subtarefas **output-only**, com **pontuação parcial**. Cada subtarefa corresponde a um valor específico de K , e você deve projetar um parque que satisfaça todas as condições acima para esse valor de K . Sua pontuação depende do número de atrações em sua solução: menos atrações resultam em uma pontuação maior ou igual.

Detalhes de Implementação

Existem dois métodos para submeter sua solução, e você pode usar qualquer um deles para cada subtarefa:

- **Chamada de procedimento**
- **Arquivo de saída**

Para submeter sua solução por meio de uma **chamada de procedimento**, você deve implementar o seguinte procedimento:

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

- K : metade do número de tipos de atrações, e também o número máximo de passarelas que podem estar conectadas a cada atração.
- Este procedimento é chamado exatamente uma vez para cada subtarefa.

O procedimento deve retornar um par (T, E) descrevendo um parque de diversões. Seja M o número de passarelas em seu parque.

- T : um vetor de comprimento N que descreve os tipos de atrações.
- E : um vetor de comprimento M descrevendo as passarelas. Para cada $0 \leq j < M$, $E[j] = (U[j], V[j])$ representa uma passarela bidirecional entre atrações distintas $U[j]$ e $V[j]$.

Para submeter sua solução por meio de um **arquivo de saída**, crie e envie um arquivo de texto no seguinte formato:

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

Observe que sua solução deve satisfazer as seguintes restrições para ser considerada **válida**:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ para cada $0 \leq i < N$, e cada tipo deve ser atribuído a pelo menos uma atração.
- $0 \leq U[j], V[j] < N$ e $U[j] \neq V[j]$ para cada $0 \leq j < M$.
- As condições 1 e 2 devem ser satisfeitas.

Restrições

- $1 \leq K \leq 50$

Pontuação

Existem 50 subtarefas, uma para cada inteiro K de 1 a 50. Para a subtarefa i ($1 \leq i \leq 50$), o valor de K é i .

Cada subtarefa tem uma pontuação S e um número alvo de atrações P de acordo com a tabela abaixo.

Subtarefas	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	$12 \cdot K$

Para cada subtarefa, se a sua solução não descrever um parque de diversões válido, a pontuação da sua solução será 0 (relatada como `Output isn't correct` no CMS).

Caso contrário, sua pontuação será calculada com base em N e nos parâmetros S e P da seguinte forma:

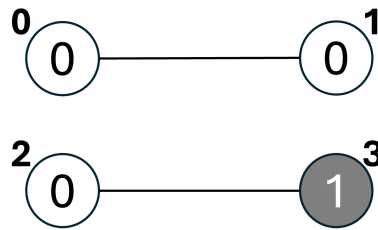
Condição	Pontuação
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Exemplo

Considere a seguinte chamada:

```
construct(1)
```

Neste exemplo, $K = 1$, então existem $2K = 2$ tipos de atrações. A figura abaixo mostra uma solução válida com $N = 4$ atrações e $M = 2$ passarelas. As atrações 0, 1, 2 são do tipo 0 e a atração 3 é do tipo 1.



Existem dois trios interessantes:

- Para a tripla de tipo $(0, 1, 0)$, podemos escolher $(a_1, a_2, a_3) = (2, 3, 2)$.
- Para a tripla de tipo $(1, 0, 1)$, podemos escolher $(a_1, a_2, a_3) = (3, 2, 3)$.

Isso significa que a Condição 1 foi satisfeita.

O procedimento pode retornar o par $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Note que a solução fornecida para este exemplo pode não ser ideal para $K = 1$.

Corretor exemplo

Formato de entrada:

K

Formato de saída:

```

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

```

Observe que a saída do corretor exemplo corresponde ao formato exigido para o arquivo de saída.