

Magic City

Thị trưởng thủ đô Tashkent muốn thiết kế lại công viên giải trí nổi tiếng Magic City. Bạn được cho một số nguyên dương K và nhiệm vụ của bạn là thiết kế một công viên như sau:

- Chọn một số nguyên N , là số lượng điểm tham quan. Các điểm tham quan được đánh số từ 0 đến $N - 1$.
- Bổ sung các lối đi bộ hai chiều, mỗi lối đi bộ nằm giữa một cặp điểm tham quan **phân biệt**. Cho phép có nhiều hơn một lối đi bộ giữa cùng một cặp điểm tham quan. **Không bắt buộc phải có khả năng di chuyển giữa mọi cặp điểm tham quan thông qua các lối đi bộ.**
- Đối với mỗi i sao cho $0 \leq i < N$, gán cho điểm tham quan i một loại $T[i]$. Có $2K$ loại điểm tham quan, được đánh số từ 0 đến $2K - 1$. Mỗi loại cần được gán cho ít nhất một điểm tham quan. Các điểm tham quan khác nhau có thể cùng thuộc một loại.

Nghiên cứu thị trường cho thấy:

- Mỗi du khách thích tham quan ba điểm tham quan và không tham quan hai điểm liên tiếp có cùng loại.
- Du khách không thích những điểm tham quan là điểm có nhiều lối đi bộ.

Để làm hài lòng **tất cả** du khách tiềm năng, thị trưởng đã đặt ra hai điều kiện đối với thiết kế của bạn.

Ta gọi một bộ ba **loại** có thứ tự (t_1, t_2, t_3) với $0 \leq t_1, t_2, t_3 < 2K$ là **thú vị** nếu $t_1 \neq t_2$ và $t_2 \neq t_3$. Lưu ý rằng t_1 có thể bằng t_3 . Do đó, có $2K \cdot (2K - 1)^2$ bộ ba thú vị.

Điều kiện 1: Với mỗi bộ ba thú vị (t_1, t_2, t_3) , phải tồn tại ba điểm tham quan a_1, a_2, a_3 (với $0 \leq a_1, a_2, a_3 < N$) sao cho:

- Các loại của a_1, a_2, a_3 tương ứng với t_1, t_2, t_3 . Tức là, $T[a_1] = t_1$, $T[a_2] = t_2$ và $T[a_3] = t_3$.
- Có một lối đi bộ giữa a_1 và a_2 .
- Có một lối đi bộ giữa a_2 và a_3 .

Việc có hay không có lối đi bộ giữa a_1 và a_3 là không quan trọng. Cũng cần lưu ý rằng khi $t_1 = t_3$, điểm tham quan a_1 và a_3 có thể giống nhau.

Điều kiện 2: Mỗi điểm tham quan chỉ có thể là điểm kết thúc của **tối đa** K lối đi bộ.

Bài toán này bao gồm 50 **subtask chỉ yêu cầu kết quả ra** với **cách chấm điểm thành phần**. Mỗi subtask tương ứng với một giá trị cụ thể của K , và bạn phải thiết kế một công viên đáp ứng tất cả các điều kiện trên cho giá trị K đó. Điểm số của bạn phụ thuộc vào số lượng điểm tham quan trong lời giải của bạn - ít điểm tham quan hơn sẽ cho điểm số cao hơn hoặc bằng.

Chi tiết cài đặt

Có hai cách để nộp lời giải, và bạn có thể sử dụng một trong hai cách cho mỗi subtask:

- **Gọi hàm**
- **File kết quả ra**

Để gửi lời giải của bạn thông qua **gọi hàm**, bạn cần cài đặt hàm sau.

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

- K : một nửa số loại điểm tham quan, và cũng là số lượng tối đa các lối đi bộ mà kết thúc ở cùng một điểm tham quan.
- Hàm này được gọi đúng một lần cho mỗi subtask.

Hàm này cần trả về một cặp (T, E) mô tả một công viên giải trí. Giả sử M là số lượng lối đi bộ trong công viên của bạn.

- T : một mảng có kích thước N mô tả loại của các điểm tham quan.
- E : một mảng có kích thước M mô tả các lối đi bộ. Với mỗi $0 \leq j < M$, $E[j] = (U[j], V[j])$ biểu thị một lối đi bộ đi hai chiều giữa các điểm tham quan phân biệt $U[j]$ và $V[j]$.

Để gửi lời giải của bạn thông qua **file kết quả ra**, hãy tạo và gửi một file văn bản theo định dạng sau:

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

Chú ý rằng lời giải của bạn phải đáp ứng các điều kiện sau để được coi là **hợp lệ**:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ với mỗi $0 \leq i < N$, và mỗi loại cần được gán cho ít nhất một điểm tham quan.
- $0 \leq U[j], V[j] < N$ và $U[j] \neq V[j]$ với mỗi $0 \leq j < M$.
- Điều kiện 1 và 2 phải thỏa mãn.

Các ràng buộc

- $1 \leq K \leq 50$

Chấm điểm

Có 50 subtask tương ứng với K từ 1 đến 50. Subtask thứ i ($1 \leq i \leq 50$), giá trị của K là i .

Mỗi subtask có một điểm số S và một số lượng điểm tham quan mong muốn P theo bảng bên dưới.

Subtask	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	$12 \cdot K$

Đối với mỗi subtask, nếu lời giải của bạn không mô tả một công viên giải trí hợp lệ, thì điểm số của lời giải sẽ là 0 (được phản hồi là `Output isn't correct` trong CMS).

Ngược lại, điểm số của bạn được tính dựa trên N và các tham số S và P như sau:

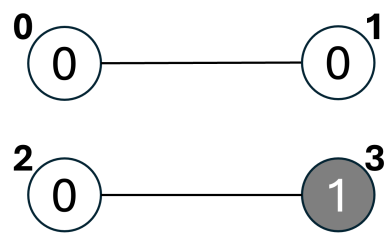
Điều kiện	Điểm
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Ví dụ

Xét lời gọi hàm sau:

```
construct(1)
```

Trong ví dụ này, $K = 1$, vậy có $2K = 2$ loại điểm tham quan. Hình dưới đây mô tả một giải pháp hợp lệ với $N = 4$ điểm tham quan và $M = 2$ lối đi bộ. Các điểm tham quan 0, 1, 2 thuộc loại 0, và điểm tham quan 3 thuộc loại 1.



Có hai bộ ba thú vị:

- Đối với bộ ba loại $(0, 1, 0)$, có thể chọn $(a_1, a_2, a_3) = (2, 3, 2)$.
- Đối với bộ ba loại $(1, 0, 1)$, có thể chọn $(a_1, a_2, a_3) = (3, 2, 3)$.

Điều này có nghĩa là Điều kiện 1 được thỏa mãn.

Hàm có thể trả về cặp $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Chú ý rằng lời giải được cung cấp cho ví dụ này có thể không tối ưu với $K = 1$.

Trình chấm mẫu

Định dạng dữ liệu vào:

```
K
```

Định dạng kết quả ra:

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

Chú ý rằng kết quả của trình chấm mẫu phù hợp với định dạng yêu cầu của file đầu ra.