Mục lục

Bài 1. Đường đi trên bảng — SUMDIS

Cho một bảng vuông $H \times W$. Ô góc trái trên có tọa độ $(0, 0)$, ô góc phải dưới có tọa độ $(H - 1, W - 1)$. Trong bảng có N ô vuông $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ được tô màu đen, các ô còn lại dược tô màu trắng.

Gọi đường đi ngắn nhất giữa hai ô trắng A và B là số bước di chuyển ngắn nhất, chỉ đi qua ô trắng, mỗi bước di chuyển có thể đi đến một ô vuông khác chung cạnh. Xét tất các các cặp không xếp thứ tự các ô trắng. Sẽ có $\binom{H \times W - N}{2}$ cặp ô trắng như vậy. Với mỗi cặp ô trắng, ta có thể tìm được đường đi ngắn nhất giữa 2 ô trắng đó.

Yêu cầu: Hãy tính tổng tất cả các độ dài đường đi ngắn nhất của các cặp ô trắng.

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dươnng H và W $(H, W \leq 10^6)$, tương ứng là số hàng và số cột của bảng;
- Dòng thứ 2 chứa một số nguyên N $(N \leq 30)$, số ô màu đen trong bảng;
- N dòng tiếp theo, dòng thứ i trong N dòng này gồm 2 số nguyên x_i và y_i là tọa độ của ô đen thứ i $(x_i < H, y_i < W)$.

  Dữ liệu đảm bảo giữa hai ô trắng luôn tồn tại 1 đường đi chỉ đi qua các ô trắng.

Kết quả

Ghi ra duy nhất một số là phần dư của tổng các khoảng cách ngắn nhất trong phép chia cho $10^9 + 7$.

Ví dụ

test	answer
2 3 1 0 1	20
4 4 1 1 1	296

Hạn chế

- Subtask 1: $H, W \leq 40$;
- Subtask 2: $1 \leq N \leq 2$;
- Subtask 3: Không có ràng buộc gì thêm.

Bài 2. Dãy cân bằng — BALANCE

Một dãy $a_1, a_2, \ldots, a_K$ được gọi là dãy cân bằng nếu thỏa mãn:

$$a_1 + a_2 + \ldots + a_K = a_1 * a_2 * \ldots * a_K.$$

Gọi $f(K)$ là số lượng dãy cân bằng có độ dài là $K (K > 1)$.

Yêu cầu: Cho một số nguyên dương N, hãy tính các giá trị $f(2), f(3), \ldots, f(N)$. Vì các giá trị rất lớn nên hãy lấy dư khi chia cho P với (P là số nguyên tố).

Dữ liệu vào

Một dòng duy nhất gồm 2 số nguyên dương N, P $(2 \le N \le 2 * 10^5, 10^8 \le P \le 10^9 + 10)$.

Kết quả

In ra trên $N - 1$ dòng các giá trị là phần dư của $f(2), f(3), \ldots, f(N)$ trong phép chia cho P.

Ví dụ

test	answer
7 804437957	1 6 12 40 30 84

Hạn chế

- Subtask 1 (20%) : $N \le 3$.
- Subtask 2 (20%) : $N \le 15$.
- Subtask 3 (20%) : $N \le 50$.
- Subtask 4 (20%) : $N \le 400$.
- Subtask 5 (20%): không giới hạn gì thêm.

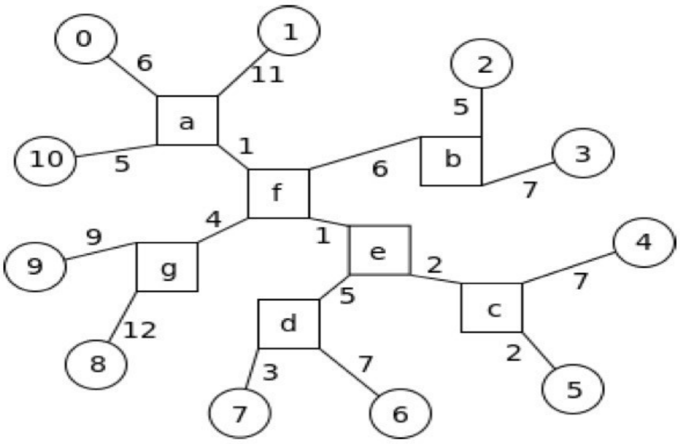
Bài 3. Thị trấn — towns

Ở đất nước K có N thị trấn nhỏ được đánh số từ 0 đến $n-1$. Ngoài ra còn có một số lượng chưa xác định các thành phố lớn. Các thị trấn nhỏ và các thành phố lớn sẽ được gọi chung là các *đô thị*.

Tất cả các đô thị của đất nước K được kết nối bởi một mạng đơn các đường cao tốc hai chiều. Mỗi đường cao tốc nối hai đô thị khác nhau và mỗi cặp đô thị được nối trực tiếp với nhau bởi nhiều nhất là một đường cao tốc. Với mỗi cặp đô thị a và b có duy nhất một đường đi mà theo đó có thể di chuyển từ a đến b sử dụng các đường cao tốc sao cho không có đường cao tốc nào được sử dụng quá một lần.

Biết rằng mỗi thị trấn nhỏ được nối trực tiếp với một đô thị khác và mỗi thành phố lớn được nối trực tiếp với ba đô thị hoặc nhiều hơn.

Hình dưới đây mô tả một mạng gồm 11 thị trấn nhỏ và 7 thành phố lớn. Các thị trấn nhỏ được khoanh bởi vòng tròn và đánh nhãn bởi số nguyên, các thành phố lớn được khoanh bởi hình vuông và đánh nhãn bởi chữ cái.



Mỗi đường cao tốc có độ dài là một số nguyên dương. Khoảng cách giữa hai đô thị là tổng nhỏ nhất các độ dài của các đường cao tốc cần đi qua để từ đô thị này đến được đô thị kia.

Với mỗi thành phố lớn C ta có thể đo khoảng cách $r(C)$ đến thị trấn ở xa thành phố lớn này nhất. Thành phố C được gọi là *hub* nếu khoảng cách $r(C)$ là nhỏ nhất trong số tất cả các thành phố. Khoảng cách giữa hub và thị trấn ở xa nó nhất sẽ được ký hiệu bởi R. Như vậy, R là số nhỏ nhất trong tất cả các giá trị $r(C)$.

Trong ví dụ nêu trên thị trấn nhỏ ở xa thành phố a nhất là thị trấn 8, và khoảng cách giữa chúng là $r(a) = 1 + 4 + 12 = 17$. Đối với thành phố g chúng ta có $r(g) = 17$. (Một trong số các thị trấn nhỏ ở cách xa thành phố nhất là thị trấn 6.) Có duy nhất một hub trong ví dụ ở trên, đó là thành phố f, với $r(f) = 16$. Do đó, trong ví dụ ở trên R bằng 16.

Việc loại bỏ một hub sẽ chia mạng ra thành một số thành phần liên thông. Một hub được gọi là *cân bằng* nếu mỗi thành phần liên thông chứa nhiều nhất $\lfloor N/2 \rfloor$ thị trấn nhỏ. (Nhấn mạnh là chúng ta không đếm số lượng thành phố lớn.) Chú ý rằng ký hiệu $\lfloor x \rfloor$ là số nguyên lớn nhất không vượt quá x.

Trong ví dụ của chúng ta, thành phố f là hub. Nếu loại bỏ thành phố f, mạng sẽ phân rã thành bốn thành phần liên thông. Bốn thành phần liên thông này bao gồm các tập thị trấn sau đây: {0,1,10}, {2,3}, {4,5,6,7}, và {8,9}. Không một thành phần nào trong số này có nhiều hơn $\lfloor 11/2 \rfloor = 5$ thị trấn nhỏ, do đó thành phố f là một hub cân bằng.

Nhiệm vụ:

Thoạt tiên, thông tin duy nhất về mạng các đô thị và các đường cao tốc mà bạn được biết chỉ là số lượng các thị trấn nhỏ N. Bạn không biết số lượng các thành phố lớn. Bạn cũng không được biết gì về bố trí của các đường cao tốc trong đất nước. Bạn chỉ có thể nhận được thông tin mới bằng việc đưa ra các truy vấn đối với khoảng cách giữa các cặp hai thị trấn nhỏ.

Nhiệm vụ của bạn là hãy xác định:

- Trong tất cả các subtask: khoảng cách R.
- Trong các subtask từ 3 đến 6: có hay chăng hub cân bằng trong mạng.

Bạn cần cài đặt hàm `hubDistance`. Chương trình chấm sẽ chấm với nhiều test ở mỗi lần chạy. Số lượng test ở mỗi lần chạy là không quá 40. Với mỗi test chương trình chấm sẽ gọi hàm `hubDistance` của bạn đúng một lần. Hãy đảm bảo là hàm của bạn phải khởi tạo tất cả các biến cần thiết mỗi khi nó được gọi. Bạn có thể xem file mẫu trong hệ thống.

`hubDistance(N, sub)`

- N: số lượng thị trấn nhỏ.
- sub: chỉ số của subtask (được giải thích trong mục Subtasks).
- Nếu sub là 1 hoặc 2, hàm cần trả lại hoặc là R hoặc là $-R$
- Nếu sub lớn hơn 2, nếu tồn tại hub cân bằng thì hàm phải trả lại R, nếu trái lại hàm phải trả lại $-R$.

Hàm `hubDistance` của bạn có thể thu được thông tin về mạng các đường cao tốc bằng cách gọi hàm `getDistance(i, j)` của chương trình chấm. Hàm này sẽ trả lại khoảng cách giữa hai thị trấn nhỏ i và j. Chú ý là nếu i và j là bằng nhau, hàm sẽ trả lại 0. Ngoài ra, hàm cũng trả lại 0 khi các đối số là không đúng đắn.

Subtasks

Trong mỗi test:

- N nằm trong khoảng giữa 6 và 110 kể cả hai đầu mút.
- Khoảng cách giữa hai thị trấn bất kỳ là nằm trong khoảng giữa 1 và 1,000,000 kể cả hai đầu mút.

Có hạn chế đối với số lượng truy vấn mà chương trình của bạn có thể đòi hỏi. Các hạn chế này là phụ thuộc vào từng subtask, và được chỉ ra trong bảng dưới đây. Nếu chương trình của bạn thực hiện vượt quá hạn chế về số truy vấn, chương trình sẽ bị ngắt và sẽ được coi là đưa ra lời giải không đúng đắn.

subtask	điểm	số lượng truy vấn	tìm hub cân bằng	ràng buộc bổ sung
1	13	$\frac{N(N-1)}{2}$	NO	không có
2	12	$\lceil\frac{7N}{2}\rceil$	NO	không có
3	13	$\frac{N(N-1)}{2}$	YES	không có
4	10	$\lceil\frac{7N}{2}\rceil$	YES	mỗi thành phố lớn được nối với đúng ba đô thị
5	13	$5N$	YES	không có
6	39	$\lceil\frac{7N}{2}\rceil$	YES	không có

Chú ý là $\lceil x\rceil$ ký hiệu số nguyên nhỏ nhất còn lớn hơn hoặc bằng x.

Ma trận kề cho ví dụ trong hình vẽ trên:

11
0 17 18 20 17 12 20 16 23 20 11
17 0 23 25 22 17 25 21 28 25 16
18 23 0 12 21 16 24 20 27 24 17
20 25 12 0 23 18 26 22 29 26 19
17 22 21 23 0 9 21 17 26 23 16
12 17 16 18 9 0 16 12 21 18 11

20 25 24 26 21 16 0 10 29 26 19
16 21 20 22 17 12 10 0 25 22 15
23 28 27 29 26 21 29 25 0 21 22
20 25 24 26 23 18 26 22 21 0 19
11 16 17 19 16 11 19 15 22 19 0