



Hello, 2528ctin_kduckp.

Chủ đề: Đồ thị cơ bản của cơ bản

Info

Rankings

Participation

Submissions

Leave contest

Thầy Trần Phùng Bính

Biểu diễn đồ thị

Submit

Time limit: 1.0s / **Memory limit:** 1G

Points: 100

Có hai cách tiêu chuẩn để biểu diễn một đồ thị, trong đó $G = (V, E)$ là tập hợp các đỉnh và E là tập hợp các cạnh; biểu diễn danh sách kề và biểu diễn ma trận kề.

Biểu diễn danh sách kề bao gồm một mảng $Adj[|V|]$ các danh sách, mỗi danh sách cho mỗi đỉnh trong $|V|$. Đối với mỗi u , danh sách kề $Adj[u]$ chứa tất cả các đỉnh v sao cho có một cạnh $(u, v) \in E$. Đó là, $Adj[u]$ bao gồm tất cả các đỉnh kề với u trong G .

Biểu diễn ma trận kề bao gồm ma trận $|V| \times |V|$ sao cho $a_{ij} = 1$ nếu $(i, j) \in E$, $a_{ij} = 0$ nếu không.

Viết một chương trình đọc một đồ thị có hướng G được biểu diễn bởi danh sách kề, và in ra biểu diễn ma trận kề của nó. G bao gồm $n (= |V|)$ đỉnh được xác định bằng chỉ số $1, 2, \dots, n$ tương ứng.

Dữ liệu

- Dòng 1: chứa một số nguyên n ;
- Tiếp theo là n dòng, dòng thứ i mô tả một danh sách các đỉnh kề với đỉnh i có dạng:
 $u \ k \ v_1 \ v_2 \ \dots \ v_k$ biểu thị danh sách gồm k đỉnh kề của đỉnh có chỉ số là u .

Kết quả

- Ghi ra biểu diễn ma trận kề của đồ thị đã cho.

Ví dụ

input	output
4	0 1 0 1
1 2 2 4	0 0 0 1
2 1 4	0 0 0 0
3 0	0 0 1 0
4 1 3	

Sample Input

Copy



Hello, 2528ctin_kduckp.

```
1 2 2 4
2 1 4
3 0
4 1 3
```

Sample Output

```
0 1 0 1
0 0 0 1
0 0 0 0
0 0 1 0
```

Copy

Kiểm tra cây

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Cho đồ thị vô hướng không trọng số gồm n đỉnh, m cạnh.

Hãy cho biết đồ thị đã cho có phải là một cây hay không.

Dữ liệu

- Dòng 1: ghi hai số nguyên n, m ($0 \leq n \leq 10^5, 0 \leq m \leq 2 \times 10^5$);
- Tiếp theo mà m dòng, mỗi dòng ghi hai số nguyên u, v ($1 \leq u, v \leq n$) mô tả một cạnh nối giữa hai đỉnh u, v .

Kết quả

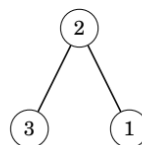
- Ghi "YES" nếu đồ thị đã cho là một cây. Ngược lại ghi "NO".

Ví dụ

input	output
3 2 1 2 2 3	YES

Giải thích ví dụ

- Hình mô tả đồ thị trong ví dụ:





Hello, 2528ctin_kduckp.

```
3 2
1 2
2 3
```

Copy

Sample Output

```
YES
```

Copy

Du lịch

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Quốc gia X có N thành phố được đánh số từ 1 đến N và M con đường được đánh số từ 1 đến M . Con đường thứ i cho phép đi từ thành phố A_i đến thành phố B_i , nhưng không cho phép đi từ thành phố B_i đến thành phố A_i .

Tèo đang lên kế hoạch cho chuyến du lịch trên quốc gia này, bắt đầu từ một thành phố nào đó, di chuyển qua không hoặc nhiều con đường, và kết thúc tại một thành phố nào đó. Hỏi có bao nhiêu cặp thành phố có thể là điểm bắt đầu và điểm kết thúc của chuyến du lịch của Tèo? Các cặp thành phố có thứ tự khác nhau cũng được coi là phân biệt.

Dữ liệu

- Dòng 1: ghi hai số nguyên N và M ($2 \leq N \leq 2000, 0 \leq M \leq \min(2000, N(N-1))$)
- Tiếp theo là M dòng, mỗi dòng ghi hai số nguyên A_i, B_i ($1 \leq A_i, B_i \leq N; A_i \neq B_i$) các cặp (A_i, B_i) đôi một phân biệt.

Kết quả

- Ghi một số nguyên duy nhất là số cặp thành phố đếm được.

Ví dụ

input	output
3 3 1 2 2 3 3 2	7

Giải thích ví dụ

- Chúng ta có bảy cặp thành phố có thể là điểm bắt đầu và điểm kết thúc: (1,1), (1,2), (1,3), (2,2), (2,3), (3,2), (3,3).



Hello, 2528ctin_kduckp.

```
3 3
1 2
2 3
3 2
```

Copy

Sample Output

```
7
```

Copy

Đếm đường đi đơn

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Cho đồ thị vô hướng không trọng số gồm n đỉnh, m cạnh. Các đỉnh và các cạnh đều được đánh số từ 1. Mỗi đỉnh của đồ thị có bậc không lớn hơn 10.

Gọi K là số lượng đường đi đơn xuất phát từ đỉnh 1 (đường đi không có cạnh lặp lại). Hãy in ra giá trị: $\min(K, 10^6)$.

Dữ liệu

- Dòng 1: ghi hai số nguyên n và m ($1 \leq n \leq 2 \times 10^5, 0 \leq m \leq \min(2 \times 10^5, \frac{n(n-1)}{2})$) tương ứng với số đỉnh và số cạnh của đồ thị.
- Tiếp theo là m dòng, dòng thứ i ghi hai số nguyên u_i, v_i ($1 \leq i \leq m$) mô tả cạnh thứ i nối hai đỉnh u_i và v_i .

Kết quả

- Ghi một số nguyên duy nhất là kết quả tìm được.

Ví dụ

input	output
4 2 1 2 2 3	3

Sample Input

Copy



Hello, 2528ctin_kduckp.

1 2
2 3

Sample Output

3

Copy

Đếm thành phần liên thông

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Cho đồ thị vô hướng không trọng số gồm n đỉnh, m cạnh. Các đỉnh và các cạnh đều được đánh số từ 1.

Hãy đếm xem trong đồ thị có bao nhiêu thành phần liên thông.

Dữ liệu

- Dòng 1: ghi hai số nguyên n và m ($1 \leq n \leq 2 \cdot 10^5, 0 \leq m \leq \frac{n(n-1)}{2}$) tương ứng với số đỉnh và số cạnh của đồ thị.
- Tiếp theo là m dòng, dòng thứ i ghi hai số nguyên u_i, v_i ($1 \leq i \leq m$) mô tả cạnh thứ i nối hai đỉnh u_i và v_i .

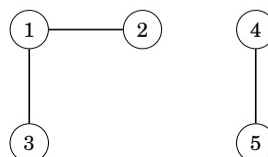
Kết quả

- Ghi một số nguyên duy nhất là số lượng thành phần liên thông của đồ thị.

Ví dụ

input	output
5 3 1 2 1 3 4 5	2
5 0	5

Giải thích ví dụ 1



Sample Input 1



Hello, 2528ctin_kduckp.

```
5 3
1 2
1 3
4 5
```

Sample Output 1

```
2
```

[Copy](#)

Sample Input 2

```
5 0
```

[Copy](#)

Sample Output 2

```
5
```

[Copy](#)

Xây dựng đường

[Submit](#)**Time limit:** 1.0s / **Memory limit:** 256M**Points:** 100



Hello, 2528ctin_kduckp.

một số con đường mới sao cho luôn có đường đi giữa mọi cặp thành phố.

Nhiệm vụ của bạn là tìm ra số lượng tối thiểu các con đường cần thiết, và xác định các con đường nào cần được xây dựng.

Dữ liệu

- Dòng đầu tiên chứa hai số nguyên n và m : số lượng thành phố và con đường. Các thành phố được đánh số từ $1, 2, \dots, n$.
- Tiếp theo là m dòng, mỗi dòng mô tả một con đường. Mỗi dòng chứa hai số nguyên a và b : có một con đường nối giữa hai thành phố a và b .
- Một con đường luôn nối hai thành phố khác nhau, và có tối đa một con đường giữa hai thành phố bất kỳ.

Kết quả

- Đầu tiên in ra một số nguyên k : số lượng con đường cần xây thêm.
- Sau đó, in ra k dòng, mỗi dòng mô tả một con đường mới. Nếu có nhiều giải pháp, bạn cần in ra giải pháp mà chỉ số thành phố được đưa ra theo thứ tự tăng dần.

Ràng buộc

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Ví dụ

Input	Output
4 2 1 2 3 4	1 2 3

Sample Input

```
4 2
1 2
3 4
```

Copy

Sample Output

```
1
2 3
```

Copy

Đếm đảo

Submit



Hello, 2528ctin_kduckp.

Points: 100

Bạn được cung cấp một bản đồ khu vực biển là một lưới các hình vuông, mỗi hình vuông đại diện cho một khu vực đất hoặc biển. Hình dưới đây là một ví dụ về một bản đồ với ô màu đen là đất, màu trắng là biển.

Bạn có thể đi từ một khu vực đất vuông này đến một khu vực đất vuông khác nếu chúng nằm kề nhau theo chiều ngang, chiều dọc hoặc đường chéo trên bản đồ. Hai khu vực được coi là cùng một hòn đảo nếu và chỉ nếu bạn có thể đi từ một khu vực này đến khu vực khác có thể thông qua các khu vực đất khác. Khu vực biển trên bản đồ được bao quanh bởi biển và do đó bạn không thể đi ra khỏi khu vực.

Bạn được yêu cầu viết một chương trình đọc bản đồ và đếm số hòn đảo trên đó. Ví dụ, bản đồ trong dưới đây bao gồm ba hòn đảo.



Dữ liệu

Dữ liệu nhập bao gồm một loạt các bộ dữ liệu, mỗi bộ dữ liệu có định dạng như sau:

- Dòng 1: ghi hai số nguyên w h ($1 \leq w, h \leq 50$);
- Tiếp theo là h dòng, mỗi dòng gồm w số nguyên mô tả quần đảo. Mỗi số nguyên chỉ có thể là 0 hoặc 1.

Kết thúc dữ liệu nhập được chỉ định bằng một dòng chứa hai số không.

Kết quả

- Đối với mỗi bộ dữ liệu, in ra số lượng các hòn đảo trên một dòng.

Ví dụ



Hello, 2528ctin_kduckp.

0	1
2 2	1
0 1	3
1 0	1
3 2	9
1 1 1	
1 1 1	
5 4	
1 0 1 0 0	
1 0 0 0 0	
1 0 1 0 1	
1 0 0 1 0	
5 4	
1 1 1 0 1	
1 0 1 0 1	
1 0 1 0 1	
1 0 1 1 1	
5 5	
1 0 1 0 1	
0 0 0 0 0	
1 0 1 0 1	
0 0 0 0 0	
1 0 1 0 1	
0 0	

Sample Input

Copy



Hello, 2528ctin_kduckp.

```
0
2 2
0 1
1 0
3 2
1 1 1
1 1 1
5 4
1 0 1 0 0
1 0 0 0 0
1 0 1 0 1
1 0 0 1 0
5 4
1 1 1 0 1
1 0 1 0 1
1 0 1 0 1
1 0 1 1 1
5 5
1 0 1 0 1
0 0 0 0 0
1 0 1 0 1
0 0 0 0 0
1 0 1 0 1
0 0
```

Sample Output

```
0
1
1
3
1
9
```

Copy

Xóa cạnh

Submit



Hello, 2528ctin_kduckp.

Points: 100

Cho đồ thị vô hướng gồm N đỉnh, M ($1 \leq N, M \leq 3000$) cạnh. Bạn cần xoá lần lượt từng đỉnh của đồ thị theo danh sách cho trước. Mỗi khi xoá một đỉnh thì các cạnh nối tới đỉnh đó sẽ bị xoá theo.

Yêu cầu: mỗi lần xoá xong một đỉnh, bạn cần cho biết lúc này đồ thị có còn liên thông hay không.

Dữ liệu

- Dòng 1: chứa hai số nguyên N và M .
- M dòng tiếp theo mô tả mỗi đường đi giữa các cặp đỉnh (đỉnh được đánh số từ 1 đến N).
- N dòng cuối cùng đưa ra một hoán vị của 1 đến N mô tả thứ tự đỉnh bị xoá.

Kết quả

Đầu ra gồm N dòng, mỗi dòng chứa "YES" hoặc "NO". Dòng đầu tiên cho biết đồ thị ban đầu có liên thông hay không, và dòng thứ $i + 1$ cho biết đồ thị có còn liên thông không sau khi xoá đỉnh thứ i đi.

Ví Dụ

input	output
4 3	YES
1 2	NO
2 3	YES
3 4	YES
3	
4	
1	
2	

Sample Input

```
4 3
1 2
2 3
3 4
3
4
1
2
```

Copy

Sample Output

Copy



Hello, 2528ctin_kduckp.

NO
YES
YES

Nhiệm vụ

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Có N người lính ($1 \leq N \leq 200$) đứng trên các điểm tọa độ nguyên của trục tọa độ Oxy . Mỗi người có cường độ âm thanh là P , nghĩa là có thể nói cho người cách mình nhiều nhất P mét nghe rõ.

Bây giờ, người chỉ huy muốn thông báo nhiệm vụ bằng cách chọn một người rồi yêu cầu người này thông báo nhiệm vụ tới những người trong phạm vi có thể nghe rõ lời của anh ta, khi một người nghe rõ nhiệm vụ, người đó cũng sẽ lặp lại thao tác tương tự.

Yêu cầu: Hãy cho biết trong trường hợp chỉ huy chọn được một người tối ưu nhất, sẽ có tối đa bao nhiêu người nhận được thông báo nhiệm vụ.

Dữ liệu

- Dòng đầu tiên chứa số N .
- N dòng tiếp theo mỗi dòng chứa ba số nguyên: tọa độ x và y của một người lính (các số nguyên trong phạm vi từ 0 đến 25,000) và P , cường độ âm thanh của người lính này.

Kết quả

Ghi ra một dòng gồm một số nguyên là số lượng người lính tối đa có thể nhận được thông báo nhiệm vụ (tính cả người ban đầu được chỉ huy chọn).

Ví Dụ

input	output
4 1 3 5 5 4 3 7 2 1 6 1 1	3

Trong ví dụ trên, nếu chỉ huy chọn người lính 1 để bắt đầu thông báo thì sẽ có số người nhận được thông báo tối đa bằng 3.

Sample Input

Copy



Hello, 2528ctin_kduckp.

```
1 3 5
5 4 3
7 2 1
6 1 1
```

Sample Output

3

Copy

Xây dựng hàng rào

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Có N ($2 \leq N \leq 10^5$) con cừu được đánh số từ 1 đến N , mỗi con cừu được đặt tại một vị trí (x, y) riêng biệt trên bản đồ 2D của trang trại. Có M cặp bò ($1 \leq M < 10^5$) là bạn của nhau. Hai con cừu mà là bạn của nhau thì thuộc cùng một mạng lưới bạn bè.

Chủ trang trại muốn xây dựng một hàng rào hình chữ nhật có các cạnh song song với các trục x và y . Ông ấy muốn đảm bảo rằng ít nhất một mạng lưới bạn bè của các con cừu được hoàn toàn bao quanh bởi hàng rào (các con cừu trên biên của hình chữ nhật được tính là đã được bao quanh).

Hãy xác định chu vi nhỏ nhất có thể của một hàng rào thỏa mãn yêu cầu này. Có thể có trường hợp hàng rào này có chiều rộng hoặc chiều cao bằng không.

Dữ liệu

- Dòng đầu tiên của đầu vào chứa N và M .
- N dòng tiếp theo mỗi dòng chứa tọa độ x và y của một con cừu (số nguyên không âm có giá trị không quá 10^8).
- M dòng tiếp theo mỗi dòng chứa hai số nguyên a và b mô tả một mối quan hệ bạn bè giữa hai con cừu a và b . Mỗi con cừu có ít nhất một mối quan hệ bạn bè, và không có mối quan hệ bạn nào được lặp lại trong dữ liệu vào.



Hello, 2528ctin_kduckp.

Ví Dụ

fenceplan.in	fenceplan.out
7 5 0 5 10 5 5 0 5 10 6 7 8 6 8 4 1 2 2 3 3 4 5 6 7 6	10

Sample Input

```
7 5
0 5
10 5
5 0
5 10
6 7
8 6
8 4
1 2
2 3
3 4
5 6
7 6
```

Copy

Sample Output

```
10
```

Copy

Đếm kim cương

Submit

Time limit: 1.0s / **Memory limit:** 256M



Hello, 2528ctin_kduckp.

Lovro có một bảng gồm n hàng và m cột, trong đó mỗi ô là ký tự chấm “.” hoặc ký tự thăng “#”. Bằng cách xoay hình vuông 45° , một hình kim cương sẽ được hình thành trên bảng. Để một phần của bảng được coi là hình thoi, cạnh của nó cũng chỉ được chứa ký tự thăng “#”, trong khi bên trong của nó phải được điền đầy đủ bằng các ký tự chấm “.” và nó phải không trống. Bên ngoài viên kim cương, bất kỳ ký tự nào cũng được phép. Kim cương có nhiều kích cỡ khác nhau và ba ví dụ nhỏ nhất về viên kim cương được hiển thị trong mẫu đầu tiên.

Viên kim cương kích thước nhỏ nhất	Viên kim cương kích thước lớn thứ 2	Viên kim cương kích thước lớn thứ 3
<pre> # # . # # </pre>	<pre> # # . # # . . . # # . # # </pre>	<pre> # # . # # . . . # # # # . . . # # . # # </pre>

Fabijan yêu cầu Lovro cho anh ta biết có bao nhiêu viên kim cương trên bàn, nếu không Lovro phải đưa cho anh ta một chiếc bánh quy. Hãy giúp Lovro bằng cách viết chương trình đếm số viên kim cương trong bảng của anh ấy.

Dữ liệu vào: Đọc từ tệp văn bản COUNT.INP có cấu trúc như sau:

Dòng đầu tiên chứa số nguyên dương n và m ($1 \leq n, m \leq 2000$), số hàng và số cột của bảng. Trong n dòng tiếp theo, mỗi dòng chứa m ký tự chấm “.” hoặc thăng “#” mô tả bảng.

Kết quả: Ghi ra tệp văn bản COUNT.OUT một giá trị duy nhất tương ứng số lượng kim cương trong bảng cho trong file dữ liệu vào.

Ví dụ:

COUNT.INP	COUNT.OUT	COUNT.INP	COUNT.OUT
<pre> 7 25 .#...#...#...#...#... #.#...#...#...#...#... .#...#...#...#...#...#...#...#...#...##...#...#...#...##...#...#...#...##...#...#...#...##...#...#...#...##...#...#...#...# </pre>	3	<pre> 11 17#.....#..#.#.....#.. ...#...#...#...# ..#...#...#...# ..#...#...#...# .#...#...#...# .#...#...#...# .#...#...#...# .#...#...#...# </pre>	1



Hello, 2528ctin_kduckp.

	#. #.....#. #.....	
COUNT.INP	COUNT.OUT		
5 11 ##. #. #. #. #. #. #. #. # . #. #. #. #. #. #. #. #. # ##. #. #. #. #	14		

Ràng buộc:

- Subtask 1: 10 test tương ứng với 1,4 điểm có $1 \leq n, m \leq 100$ và bảng chỉ chứa các viên kim cương có kích thước nhỏ nhất;
- Subtask 2: 20 test tương ứng với 2,8 điểm có $1 \leq n, m \leq 100$;
- Subtask 3: 20 test tương ứng với 2,8 điểm có $100 \leq n, m \leq 2000$

Sample Input 1

7 25

. # . . # . . . # . . . # . . .
. # . # . . . # . # . . . # . # . .
. # . . # . # . . . # . # . . . # .
. . . . # . . . # . # . . . # #
. . . . # . . . # . . . # . . . # .
. . . . # # . . . # . # . .
. . . . # # . . . # . . .

Copy

Sample Output 1

3

Copy

Sample Input 2

Copy



Hello, 2528ctin_kduckp.

```
.....#.....#..
....#.#.....#.
...#...#...#...#
..#.....#...#.#.
.#.....#...#...#..
#....#.#..#.....
.#.....#..#.....
..#.....#.....
...#...#.....
....#.#.....
.....#.....
```

Sample Output 2

1

Copy

Sample Input 3

```
5 11
##.##.##.
#.#.#.##.
.#.##.##.
#.#.#.##.
##.##.##.
```

Copy

Sample Output 3

14

Copy

Các miền trên bảng

Submit



Hello, 2528ctin_kduckp.

Points: 100

Cho một bảng chữ nhật được chia thành $M \times N$ ô vuông (M dòng, N cột). Mỗi ô vuông ghi một số nguyên dương (trong khoảng từ 1 đến 255). Một miền của bảng là tập hợp tất cả các ô có cùng giá trị số sao cho chúng đi được sang nhau bằng cách đi qua các ô có chung cạnh và có cùng giá trị số đang xét. Địa chỉ của một miền là tọa độ [dòng, cột] của ô đầu tiên thuộc miền theo thứ tự duyệt từ trái sang phải và từ trên xuống dưới. Diện tích của một miền là số ô thuộc miền đó. *Ví dụ: Bảng dưới đây có 5 miền, miền tô màu xám (giá trị các ô là 2) có địa chỉ là [1, 3] và diện tích là 8.*

1	1	2	2	2
1	2	2	1	2
3	1	2	1	2

Yêu cầu: Xác định số miền của bảng và tìm miền có diện tích lớn nhất (chỉ rõ giá trị diện tích và địa chỉ của miền)

Dữ liệu: Vào từ file văn bản MIEN.INP có dạng:

```
M N
A[1, 1] A[1, 2] .....A[1, N]
A[2, 1] A[2, 2] .....A[2, N]
.....
A[M, 1] A[M, 2] .....A[M, N]
```

trong đó $A[i, j]$ là giá trị số của ô $[i, j]$, các số trên cùng một dòng ghi cách nhau ít nhất một dấu trắng. Giới hạn: $M, N < 101$

Kết quả: Ghi ra file MIEN.OUT

- +) Dòng đầu tiên ghi số miền của bảng.
- +) Dòng thứ hai ghi 3 số S, x, y , trong đó S là diện tích còn $[x, y]$ là địa chỉ của miền lớn nhất.

Ví dụ:

MIEN.INP	MIEN.OUT
3 5	5
1 1 2 2 2	8 1 3
1 2 2 1 2	
3 1 2 1 2	

Sample Input

```
3 5
1 1 2 2 2
1 2 2 1 2
3 1 2 1 2
```

Copy

Sample Output

Copy



Hello, 2528ctin_kduckp.

✓

8 1 3

Bộ sưu tập có giá trị

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Một bộ sưu tập tiền xu cổ được coi là có giá trị phải gồm không ít hơn Z_0 đồng tiền vàng, S_0 đồng tiền bạc và M_0 đồng tiền đồng. Bộ sưu tập ban đầu của Alibaba có một số lượng nhất định các đồng tiền vàng bạc và đồng nhưng chưa phải là một bộ sưu tập có giá trị. Tại Trụ sở của hiệp hội những người sưu tập tiền cổ có đặt một máy đổi tiền để giúp hội viên đổi được các bộ sưu tập có giá trị. Tuy nhiên, máy đổi tiền chỉ hỗ trợ việc đổi tiền trọn gói theo qui tắc đổi gói (Z_1, S_1, M_1) lấy gói (Z_2, S_2, M_2) đồng tiền. Các qui tắc đổi tiền khác nhau từng đôi một, được gán số hiệu tuần tự $1, 2, 3, \dots$ và được công bố trước. Hội viên có thể tạo gói tiền thích hợp từ bộ sưu tập của mình để thực hiện việc đổi tiền. Số lần đổi tiền là không hạn chế, tuy nhiên, để ngăn chặn việc đầu cơ, Hiệp hội qui định mỗi hội viên không được giữ quá 4 đồng tiền mỗi loại. Các đồng tiền nhận được sau mỗi lần đổi được gộp lại với các đồng tiền mà hội viên đang có để thành *một bộ sưu tập mới* và có thể được sử dụng để đổi trong những lần sau nếu cần.

Yêu cầu: Cho biết số lượng Z, S, M các đồng tiền vàng, bạc, đồng mà Alibaba có ban đầu và các qui tắc đổi tiền. Hãy cho biết Alibaba cần sử dụng ít nhất bao nhiêu qui tắc đổi tiền để có được bộ sưu tập có giá trị. Dữ liệu vào đảm bảo luôn có phương án.

Dữ liệu: Vào từ file văn bản COLLECT.INP :

+ Dòng đầu ghi 6 số nguyên không âm Z, S, M, Z_0, S_0, M_0 ($0 \leq Z, S, M, Z_0, S_0, M_0 \leq 4$);

+ Các dòng tiếp theo mỗi dòng ghi 6 số nguyên không âm $Z_1, S_1, M_1, Z_2, S_2, M_2$ xác định một qui tắc đổi tiền.

Kết quả: Đưa ra file văn bản COLLECT.OUT một dòng ghi số lần đổi tiền ít nhất để có được bộ sưu tập có giá trị.

Ví dụ:

COLLECT.INP	COLLECT.OUT
4 0 1 3 3 3	2
1 0 1 0 2 2	Giải thích: sử dụng qui tắc 3 và 4
0 1 1 0 0 3	
2 0 1 1 2 3	
1 0 0 1 1 0	

Sample Input

Copy



Hello, 2528ctin_kduckp.

```
1 0 1 0 2 2
0 1 1 0 0 3
2 0 1 1 2 3
1 0 0 1 1 0
```

Sample Output

2

Copy

Thỏ và cà rốt Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Trong một mảnh vườn hình chữ nhật có cạnh là m và n , người ta trồng cà rốt trong những ô đơn vị hình vuông có cạnh bằng 1. Trong mảnh vườn này có một chú thỏ ở trong một hàng chiếm diện tích 1 ô vuông đơn vị, chú thỏ này cần xác định miền người ta đã trồng cà rốt có diện tích lớn nhất trong mảnh vườn để đào một đường hầm ngắn nhất theo phương dọc hoặc phương ngang từ hàng đến phần diện tích lớn nhất đó. Hai ô thuộc cùng một miền có giá trị bằng 1 và đi sang nhau được qua các ô kề cạnh cũng có giá trị bằng 1. Hai miền khác nhau không có một ô vuông nào chung cạnh.

Dữ liệu: Đọc từ file văn bản CAROT.INP có cấu trúc:

+ Dòng đầu tiên ghi 4 số m, n, x, y tương ứng là kích thước mảnh vườn và vị trí hàng thỏ trong mảnh vườn ($1 \leq m, n \leq 200$).

+ Trong m dòng tiếp theo, mỗi dòng có n số 0 hoặc 1 thể hiện hàng thứ i của mảnh vườn với ý nghĩa 0 là không trồng cà rốt, 1 là có trồng cà rốt.

Kết quả: Ghi ra file CAROT.OUT gồm 2 số S và d tương ứng là diện tích miền trồng cà rốt lớn nhất và chiều dài đường hầm ngắn nhất tìm được. Đương nhiên là nếu có nhiều miền trồng cà rốt lớn nhất thì thỏ sẽ chọn miền có d nhỏ nhất. Nếu thỏ đang ở trong miền trồng cà rốt lớn nhất thì $d=0$.

Ví dụ:

CAROT.INP	CAROT.OUT
6 6 1 1	10 4
0 0 0 0 1 1	
0 0 0 0 1 1	
0 0 0 0 1 1	
0 0 0 0 1 1	

Sample Input



Hello, 2528ctin_kduckp.

```
0 0 0 0 1 1
0 0 0 0 1 1
0 0 0 0 1 1
0 0 0 0 1 1
0 0 0 0 1 1
1 1 1 0 0 0
```

Sample Output

10 4

Copy

Hệ thống xe Bus

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Hệ thống xe bus của thành phố nơi cậu sinh viên Z đang ở là một đồ thị vô hướng gồm có n trạm và m tuyến đường. Thành phố của cậu đang có kế hoạch bỏ đi một trạm và tuyến đường đã cũ và xây mới trong tương lai. Là một người thường xuyên sử dụng các phương tiện công cộng như xe bus để bảo vệ môi trường, Z nhận thấy một số trạm và tuyến đường khi nó bị gỡ bỏ sẽ lập tức ảnh hưởng tới việc đi lại.

Hãy giúp Z đếm số trạm và số tuyến đường nếu bị gỡ bỏ sẽ có thể làm một vài cặp trạm không thể đi lại được như trước, một cặp trạm được gọi là không thể đi lại được như trước nếu như trước khi gỡ bỏ từ u có thể đi được với v nhưng sau khi bỏ đi một trạm hoặc tuyến đường nào đó thì u không thể đi được tới v nữa.

Dữ liệu

- Dòng đầu tiên chứa hai số nguyên dương n, m ($1 < n < 10^4, 1 < m < 10^5$) lần lượt là số trạm và số tuyến đường.
- m dòng tiếp theo, mỗi dòng chứa hai số nguyên dương u, v ($1 < u, v < n$) là hai trạm xe bus có đường đi trực tiếp tới nhau.

Kết quả

- Gồm một dòng duy nhất chứa hai số nguyên dương lần lượt là số trạm và số tuyến đường nếu bỏ sẽ ngay lập tức gây ảnh hưởng tới việc đi lại.

Giới hạn:

Subtask 1: (25% số điểm) $n, m \leq 10^3$.

Subtask 2: (75% số điểm) Không có ràng buộc gì thêm.



Hello, 2528ctin_kduckp.

```
12 13
1 9
3 1
11 12
5 11
12 7
4 12
3 8
6 12
10 7
4 3
10 1
10 6
2 9
```

Copy

Sample Output

```
5 5
```

Copy

COVGAM

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Nói bộ não loài người càng bị đẩy đến cực hạn thì càng tiến hóa quả không sai. Mới đây, đứng trước tốc độ gia tăng chóng mặt số ca lây nhiễm cũng như tử vong của đại dịch Covid-19, một nhà khoa học thiên tài, tiến sĩ X, đã phát minh ra một phương pháp gọi là Tia gamma trùng sinh. Phương pháp này sử dụng tia gamma mang năng lượng cực lớn tác động lên 1 tế bào virus, vì năng lượng mà tế bào này hấp thụ quá lớn dẫn đến tính năng sinh lý của tế bào bị đảo ngược, biến virus trở thành một kháng thể, chiến đấu chống lại virus. Triết học võ thuật



Hello, 2528ctin_kduckp.

Tuy vậy, phương pháp này không khả thi vì cơ thể người bình thường chỉ có thể chịu được tối đa 1 phát bắn tia gamma trùng sinh này, nếu quá liều sẽ gây hậu quả nghiêm trọng đến tính mạng. Tiến sĩ X tiếp tục nghiên cứu và phát hiện ra cơ thể người vốn đã có sẵn một số lượng kháng thể, và các kháng thể này chỉ tiêu diệt được virus nếu chúng được xếp cạnh nhau tạo thành 1 đội quân kháng thể, đội quân này càng lớn thì hiệu quả càng cao. Vì vậy, việc cần làm bây giờ là phải chọn bắn tia gamma vào tế bào virus nào để đạt được hiệu quả điều trị cao nhất.

Các tế bào người bệnh được trích xuất và mô tả dưới dạng một bảng $n \times m$ ô, mỗi ô đại diện cho 1 tế bào. Các ô được biểu diễn bởi số 1 nếu nó là tế bào kháng thể, số 0 nếu nó là tế bào virus. Các tế bào kháng thể (các ô số 1) nằm cạnh nhau (không tính đường chéo) sẽ tạo thành đội quân kháng thể. Một phát bắn gamma trùng sinh sẽ biến một tế bào virus bất kì thành tế bào kháng thể.

Yêu cầu: Cho sơ đồ tế bào, tính kích thước đội quân kháng thể lớn nhất có thể tạo được bằng cách bắn tối đa 1 tia gamma trùng sinh (có thể không bắn).

Dữ liệu

- Dòng đầu tiên chứa hai số nguyên dương lần lượt là n, m ($n, m < 10^3$).
- n dòng sau đó, mỗi dòng chứa m số nguyên 0 hoặc 1 mang giá trị các tế bào.

Kết quả

- Gồm một dòng duy nhất chứa kết quả của bài toán.

Ví dụ

COVGAM.INP	COVGAM.OUT
<pre> 5 5 1 0 1 0 1 1 1 1 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 </pre>	<pre> 12 </pre>

Giải thích

Bắn tia gamma vào tế bào ở vị trí hàng 2 cột 4.

Sample Input

```

5 5
1 0 1 0 1
1 1 1 0 1
1 0 0 0 0
1 1 1 0 0
0 0 0 1 0
                    
```

Copy

Sample Output

```

12
                    
```

Copy



Hello, 2528ctin_kduckp.

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Cho một bàn cờ vua kích thước $n \times n$. Các ô được đánh số từ 1 đến n từ trên xuống dưới và từ 1 đến n từ trái sang phải.

Alice và Bobs rủ nhau chơi cờ vua trên bàn cờ này. Luật của bàn cờ này như sau:

- Alice sẽ chơi bên cờ trắng và chỉ có duy nhất 1 quân hậu ở ô (a_x, a_y) . Trong suốt trận đấu, Alice sẽ không được di chuyển quân hậu này.
- Bobs sẽ chơi bên cờ đen và chỉ có duy nhất 1 quân vua ở ô (b_x, b_y) . Sau mỗi lượt, Bobs được di chuyển quân vua theo luật cờ vua bình thường nhưng không được di chuyển vào ô bị quân hậu chiếu.
- Nếu sau một số lượt chơi nhất định, Bobs có thể đưa quân vua tới ô (c_x, c_y) trên bàn cờ thì Bobs thắng và ngược lại Alice thắng.

Alice và Bobs sẽ chơi t ván cờ. Ở ván cờ thứ i , nếu Alice thắng, in ra kí tự A, nếu Bobs thắng, in ra kí tự B. Hãy giúp đôi bạn này nhé!

Dữ liệu:

- Dòng đầu tiên chứa một số nguyên dương t là số bộ test ($t < 10^5$).
- Tiếp theo đó là t bộ test, mỗi bộ test chứa 4 dòng như sau:
 - Dòng thứ nhất chứa một số nguyên dương là n ($n < 10^{18}$).
 - Dòng thứ hai chứa hai số nguyên dương lần lượt là a_x, a_y ($a_x, a_y < n$).
 - Dòng thứ ba chứa hai số nguyên dương lần lượt là b_x, b_y ($b_x, b_y < n$).
 - Dòng thứ hai chứa hai số nguyên dương lần lượt là c_x, c_y ($c_x, c_y < n$).

Kết quả:

- Gồm t dòng, dòng thứ i là kí tự A hoặc B là đáp án của test thứ i .

Ví dụ

KESCAPE.INP	KESCAPE.OUT
3	B
8	A
4 4	A
1 3	
3 1	
8	
4 4	
2 3	
1 6	
8	
3 5	
1 2	
6 1	

Giải thích:

Ở test thứ nhất, Bobs có thể di chuyển quân vua từ ô $(1, 3) \wedge (1, 2) \wedge (2, 1) \wedge (3, 1)$ nên Bobs chiến thắng.

Giới hạn:

Subtask 1 (25% số điểm): $t = 1, n < 10^3$.

Subtask 2 (75% số điểm): Không có ràng buộc gì thêm.

Sample Input

Copy



Hello, 2528ctin_kduckp.

```
8
4 4
1 3
3 1
8
4 4
2 3
1 6
8
3 5
1 2
6 1
```

Sample Output

```
B
A
A
```

Copy

MDIV

Submit

Time limit: 1.0s / **Memory limit:** 256M

Points: 100

Clary là một người có trí nhớ rất tốt. Đối với mọi kì thi anh ta đã trải qua, anh ta đều nhớ hết đề bài và điểm số của mình. Một hôm Clary vô tình nhớ lại bài tập trong kì thi rất quan trọng đối với anh ấy vào 3 năm trước. Đề bài của bài ấy như sau:

Cho n chữ số ($n < 10$) và một số nguyên dương k ($k < 5 \times 10^5$). Hãy tìm số nguyên dương bé nhất chia hết cho k mà các chữ số của chúng chỉ nằm trong n chữ số đã cho. Nếu không tìm



Hello, 2528ctin_kduckp.

- Dòng đầu tiên chứa 2 số nguyên dương lần lượt là n, k ($n < 10$; $k < 5 \times 10^2$).
- Dòng thứ hai chứa n chữ số.

Kết quả:

- Gồm một số nguyên duy nhất là kết quả của bài toán.

Ví dụ:

MDIV.INP	MDIV.OUT
2 12 1 4	144
3 13 2 5 7	52

Giới hạn:

Subtask 1 (25% số điểm): Đáp án luôn $< 10^7$.

Subtask 2 (75% số điểm): Không có ràng buộc gì thêm.

Sample Input 1

```
2 12
1 4
```

Copy

Sample Output 1

```
144
```

Copy

Sample Input 2

```
3 13
2 5 7
```

Copy

Sample Output 2

```
52
```

Copy

STABLE

Submit



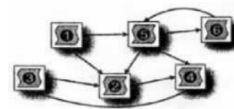
Hello, 2528ctin_kduckp.

Points: 100

Trong mạng xã hội, mỗi trang web được tổ chức trên một máy tính thành viên và cung cấp dịch vụ truy nhập tới một số trang web khác. Để truy nhập tới một trang web nào đó không có trong danh mục kết nối trực tiếp của mình, người dùng phải truy nhập tới trang web khác có kết nối với mình, dựa vào danh mục dịch vụ của trang web này để chuyển tới trang web khác theo tùy chọn, cứ như thế cho đến khi tới được trang web mình cần. Thời gian để truy nhập tới một trang web phụ thuộc chủ yếu và số lần mở trang web trong quá trình truy nhập. Như vậy, người dùng cần chủ động chọn lộ trình truy nhập hợp lí.

Sau một thời gian làm việc trên mạng, Sáng - một thành viên nhiệt thành đã tích lũy kinh nghiệm, tạo một cơ sở dữ liệu, cho biết từ một trang web có thể đi tới những trang web nào trong mạng. Trong cơ sở dữ liệu, các trang web được đánh số từ 1 đến n và có m bản ghi, mỗi bản ghi có dạng cặp có thứ tự (u, v) cho biết trang web u có kết nối tới trang web v ($1 < u, v < n, u \neq v$). Cơ sở dữ liệu chưa được chuẩn hóa, vì vậy có thể chứa các cặp (u, v) giống nhau.

Trang web của Sáng có số hiệu là s . Dựa vào cơ sở dữ liệu, Sáng có thể xác định lộ trình truy nhập nhanh nhất (tức là số lần phải mở trang web là ít nhất) từ trang web s tới trang web u bất kì. Tuy vậy, ở mạng xã hội, mọi chuyện đều có thể xảy ra: một khu vực nào đó bị mất điện, máy của một thành viên bị hỏng, trang web đó đang bị đóng để nâng cấp, ... Kết quả là một vài trang web nào đó có thể tạm thời không hoạt động. Như vậy, nếu từ s có ít nhất hai lộ trình nhanh nhất khác nhau tới u thì khả năng thực hiện truy nhập được một cách nhanh nhất tới u là lớn hơn so với những trang web chỉ có duy nhất một lộ trình nhanh nhất. Hai lộ trình gọi là khác nhau nếu có ít nhất một trang web có ở lộ trình này mà không có ở lộ trình kia hoặc cả hai lộ trình cùng đi qua những trang web như nhau nhưng theo các trình tự khác nhau. Những trang web mà từ s tới đó có ít ra là hai lộ trình nhanh nhất khác nhau được gọi là ổn định đối với s . Trang web mà từ s không có lộ trình tới nó là không ổn định đối với s .



Ví dụ, với mạng nêu ở hình bên ($n = 6, m = 9$) các trang web 4 và 3 là ổn định với $s = 1$ (từ 1 tới 4 có 2 lộ trình nhanh nhất: $1 - 2 - 4$ và $1 - 5 - 4$, từ 1 tới 3 cũng có 2 lộ trình nhanh nhất: $1 - 2 - 4 - 3$ và $1 - 5 - 4 - 3$).

Dữ liệu:

- Dòng đầu tiên chứa 3 số nguyên dương lần lượt là n, m, s ($n < 10^4, m < 5 \times 10^4, s < n$).
- m dòng tiếp theo, mỗi dòng chứa hai số nguyên dương u, v ($u, v < n$).

Kết quả:

- Gồm một số duy nhất là số trang web ổn định đối với s .

Ví dụ

STABLE.INP	STABLE.OUT
<pre> 6 11 1 1 5 1 5 5 6 1 2 5 4 2 4 4 3 5 4 5 2 3 2 6 5 </pre>	<pre> 2 </pre>

[Link submit vnoi](#)



Hello, 2528ctin_kduckp.

```
6 11 1
1 5
1 5
5 6
1 2
5 4
2 4
4 3
5 4
5 2
3 2
6 5
```

Copy

Sample Output

```
2
```

Copy