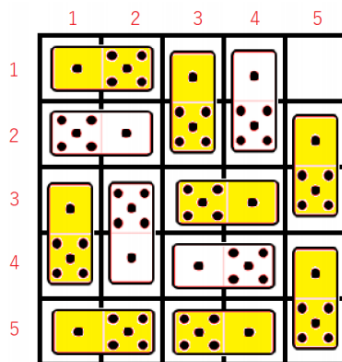


Bài A. DOMINO

File dữ liệu vào: DOMINO.inp
File kết quả: DOMINO.out
Hạn chế thời gian: 1s
Hạn chế bộ nhớ: 512M

Bờm và Cuội gần đây đã tìm thấy một bảng trò chơi xếp hình domino cổ đại. Bảng có dạng hình chữ nhật được chia thành các lưới ô vuông kích thước $N \times M$, các hàng được đánh số từ 1 đến N , các cột được đánh số từ 1 đến M , và N và M là các số tự nhiên lẻ. Bảng được lấp đầy bởi $\frac{N \times M - 1}{2}$ quân domino, mỗi domino chiếm hai ô liền kề, một số domino được đặt theo chiều ngang, một số lại được đặt theo chiều dọc, và rõ ràng sẽ có một ô trên bảng không bị quân domino nào đặt lên.

Bờm và Cuội nhìn vào bảng này và nghĩ về tất cả những nhiệm vụ tuyệt vời có thể nghĩ ra từ bảng xếp hình domino này. Một ý tưởng ngay lập tức nảy ra với họ. Với cách đặt các ô domino trên một bảng hiện tại, thì có thể di chuyển được bao nhiêu quân domino từ vị trí ban đầu của chúng đến vị trí khác, hay có bao nhiêu quân domino có thể thay đổi vị trí?



Ví dụ, đối với một bảng được sắp xếp như trên, ô trống ở vị trí (1,5), trước tiên chúng ta có thể di chuyển domino đặt trên các ô (2,5), (3,5) lên trên và do đó, ô trống sẽ là ô (3,5). Sau khi di chuyển nó, nhiều khả năng hơn sẽ mở ra, vì vậy giả sử ta có thể di chuyển domino ở các ô (4,5), (5,5) lên trên hoặc domino ở (3,3), (3,4) về phía phải. Trong tổng số 12 quân cờ domino trên bảng này, tám quân trong đó có thể được di chuyển khỏi vị trí ban đầu của chúng theo một cách nào đó.

Hãy viết một chương trình đếm xem có bao nhiêu quân cờ domino khác nhau có thể được di chuyển ra vị trí khác theo một cách nào đó.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên N và M là số hàng và số cột của bảng, cả hai đều là các số lẻ ($1 \leq N, M < 500$).
- Dòng thứ i trong $\frac{N \times M - 1}{2}$ tiếp theo, mỗi dòng chứa bốn số nguyên dương x_1, y_1, x_2, y_2 tương ứng là vị trí đặt của quân domino thứ i trên bảng là hai ô (x_1, y_1) và (x_2, y_2) , với x_1, x_2 là chỉ số hàng và y_1, y_2 là chỉ số cột. Dữ liệu đảm bảo các quân domino không xếp chồng lên nhau, không vượt quá giới hạn của bảng, và hai ô của một quân domino đặt lên là hai ô kề nhau trên bảng.

Kết quả

- Ghi ra một số nguyên duy nhất là số quân domino có thể di chuyển ra vị trí khác với vị trí ban đầu.

Ví dụ

DOMINO.inp	DOMINO.out
5 5 4 3 4 4 4 5 5 5 5 2 5 1 1 2 1 1 3 4 3 3 5 4 5 3 4 1 3 1 3 2 4 2 2 3 1 3 2 5 3 5 1 4 2 4 2 1 2 2	8

Hạn chế

- Subtask 1: 45% số điểm có $1 \leq N, M \leq 5$
- Subtask 2: 25% số điểm có $N = M = 9$, và tối đa có 2 quân domino có thể di chuyển
- Subtask 3: 30% số điểm còn lại không có ràng buộc gì thêm.

Bài B. GENTEST

File dữ liệu vào: GENTEST.inp
File kết quả: GENTEST.out
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 512M

Ban ra đề đang viết chương trình sinh test cho một bài về đồ thị. Họ cần sinh một đơn đồ thị vô hướng n đỉnh m cạnh. Thuật toán sinh như sau:

1. Nếu đã lấy đủ m cạnh thì dừng.
2. Gọi E là tập các cặp (u, v) với $u < v$ và u đang không kề với v , tức E là tập các cạnh chưa có trong đồ thị.
3. Sắp xếp E theo thứ tự từ điển $((u, v)$ đứng trước (x, y) nếu hoặc là $u < x$ hoặc là $u = x$ và $v < y$).
4. Chọn một số tự nhiên i trong phạm vi từ 1 đến $|E|$.
5. Nạp cạnh thứ i trong E vào đồ thị, lặp lại bước 1.

Cho biết các số được chọn ở bước thứ 4, hãy giúp ban ra đề in ra các cạnh của đồ thị. Lưu ý, các loại chỉ số trong bài đều được đánh số bắt đầu từ 1.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương là số đỉnh và số cạnh của đồ thị: $n\ m$
- m dòng tiếp theo, dòng thứ t chứa một số nguyên dương là số được chọn ở bước thứ 4 của lần lặp thứ t : i_t

Kết quả

In ra m cạnh được chọn theo thứ tự chọn của thuật toán. Với mỗi cạnh, in ra hai đỉnh trên một dòng cách nhau bởi dấu cách, đỉnh bé hơn phải được in trước.

Ví dụ

GENTEST.inp	GENTEST.out
4 4	1 4
3 3 1 3	2 3
	1 2
	3 4

Giải thích

- Ban đầu $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$ nên cạnh thứ 3 là $(1, 4)$
- Tiếp theo $E = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)\}$ nên cạnh thứ 3 là $(2, 3)$
- Tiếp theo $E = \{(1, 2), (1, 3), (2, 4), (3, 4)\}$ nên cạnh thứ 1 là $(1, 2)$
- Tiếp theo $E = \{(1, 3), (2, 4), (3, 4)\}$ nên cạnh thứ 3 là $(3, 4)$

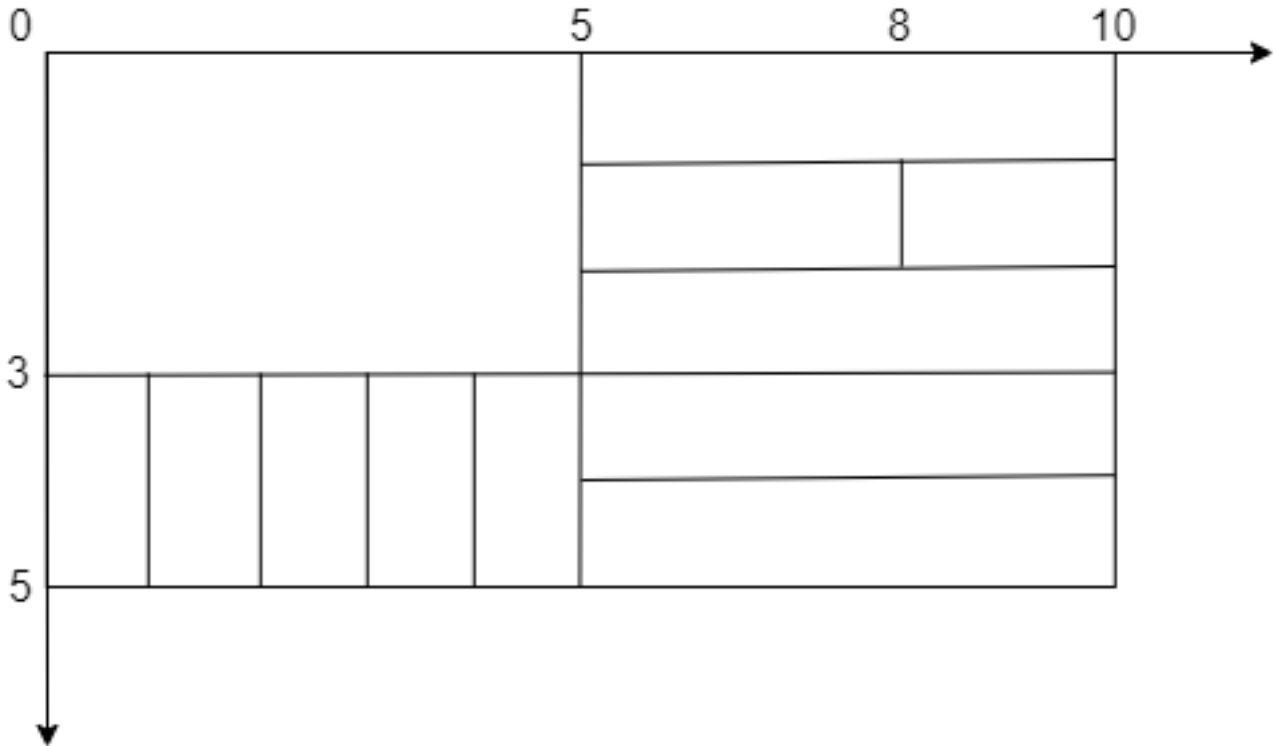
Hạn chế

- Trong tất cả các test: $1 \leq n \leq 10^9, 1 \leq m \leq 10^5$
- Có 10% số test với: $1 \leq n, m \leq 500$
- 30% test tiếp theo với: $1 \leq m \leq 500$
- 30% test tiếp theo với: $1 \leq n \leq 500$
- 30% test tiếp theo với ràng buộc gốc

Bài C. WOOD

File dữ liệu vào: `WOOD.inp`
File kết quả: `WOOD.out`
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 512M

Hùng vừa cắt một tấm gỗ mỏng hình chữ nhật thành nhiều miếng hình chữ nhật. Máy cưa cưa dùng mỗi lần sẽ nhận vào một miếng gỗ hình chữ nhật và cắt làm hai hình chữ nhật với tổng diện tích không đổi. Vị trí cắt có thể điều chỉnh được, chi phí cắt phụ thuộc vào trọng lượng miếng gỗ, do miếng gỗ ban đầu có độ dày đồng đều nên chi phí cắt có thể xem là diện tích miếng gỗ đó.



Miếng gỗ ban đầu có kích thước $n \times m$. Hùng đã dùng bút vẽ $n - 1$ đường kẻ ngang và $m - 1$ đường kẻ dọc, kết hợp với các đường biên chia hình chữ nhật thành một lưới tọa độ. Các điểm được đánh tọa độ từ $(0, 0)$ đến (n, m) từ trên xuống dưới, từ trái qua phải. Các lát cắt đều song song với biên và được đánh dấu bởi tọa độ các điểm này. Mỗi lát cắt là một đường liên tục và cắt đứt điểm một miếng nào đó làm hai phần. Miếng gỗ ban đầu đã được cắt xong, Hùng cần báo cáo lại quá trình cắt. Cậu nhớ chính xác tọa độ tất cả các lát cắt mà mình thực hiện, tuy nhiên không nhớ chính xác thứ tự đã cắt. Hùng đưa cho bạn danh sách các lát cắt này và nhờ bạn sắp xếp lại thứ tự cho hợp lý. Cậu cũng biết rằng có thể có nhiều thứ tự khác nhau, và quả quyết rằng trong số các thứ tự đúng đó, mình đã cắt theo một thứ có tổng chi phí cắt là nhỏ nhất.

Yêu cầu: Hãy giúp Hùng tìm ra một thứ tự cắt (tức hoán vị các lát cắt theo trí nhớ của Hùng) hợp lý và có tổng chi phí nhỏ nhất, hoặc thông báo là cậu đã nhớ nhầm.

Dữ liệu vào

- Dòng đầu chứa ba số nguyên dương là kích thước ban đầu và số lát cắt: $n \ m \ k$ ($1 \leq n, m \leq 10^9, 1 \leq k \leq 10^5$)
- k dòng tiếp theo, mỗi dòng ghi một lát cắt: $u \ v \ x \ y$ ($0 \leq u \leq x \leq n, 0 \leq v \leq y \leq m$) có nghĩa lát cắt này kéo dài từ điểm (u, v) đến điểm (x, y) . Dữ liệu đảm bảo lát cắt song song với biên, không trùng lên biên của hình chữ nhật đang cắt và có độ dài khác 0.

Kết quả

Nếu không có thứ tự nào hợp lý, in ra -1. Ngược lại in ra trên hai dòng:

- Dòng đầu chứa chỉ phí cắt nhỏ nhất tìm được
- Dòng tiếp theo chứa k số là một hoán vị của $\{1, 2, \dots, k\}$ cho biết thứ tự các lát cắt tìm được.

Dữ liệu đảm bảo chi phí nhỏ nhất nếu có sẽ không vượt quá 10^{18}

Ví dụ

WOOD.inp	WOOD.out
5 10 11 3 5 5 5 3 0 3 10 0 5 3 5 1 5 1 10 2 5 2 10 4 5 4 10 3 1 5 1 3 2 5 2 3 3 5 3 3 4 5 4 1 8 2 8	164 2 1 3 4 5 6 8 7 9 11 10

Hạn chế

Với mỗi test, nếu chỉ tính đúng chi phí cắt sẽ được 50% điểm của test đó, để nhận được 50% điểm này thì dòng thứ hai bạn cần để trống.

- Subtask 1 (20%): $k \leq 8$
- Subtask 2 (20%): $n = 1$
- Subtask 3 (20%): Thứ tự Hùng nhớ là một thứ tự hợp lý và tối ưu
- Subtask 4 (20%): Thứ tự Hùng nhớ là một thứ tự hợp lý
- Subtask 5 (20%): Ràng buộc gốc