
ĐƯỜNG ĐI DÀI NHẤT (MAXPATH.*)

(Bài tập cơ bản phần đồ thị)

Cho đồ thị có hướng, không có chu trình (DAG). Mỗi cạnh của đồ thị được gán với một số nguyên gọi là trọng số của cạnh. Trọng số của một đường đi là tổng trọng số các cạnh trên đường đi đó.

Hãy tìm một đường đi có trọng số lớn nhất từ đỉnh s đến đỉnh t .

Dữ liệu: Vào từ file văn bản MAXPATH.INP

- Dòng đầu tiên ghi 4 số nguyên dương n, m, s, t ($n \leq 5 \cdot 10^4, m \leq 10^5$) lần lượt là số đỉnh, số cung và số hiệu đỉnh xuất phát, kết thúc
- m dòng tiếp theo, mỗi dòng ghi ba số u, v, w thể hiện có một cung nối đỉnh u với đỉnh v có trọng số w ($1 \leq u, v \leq n; |w| \leq 10^3$)

Kết quả: Ghi ra file văn bản MAXPATH.OUT

- Dòng đầu tiên ghi YES/NO tùy theo có/không có đường đi từ s đến t .
- Nếu dòng đầu ghi YES thì dòng thứ hai ghi K là trọng số của đường đi tìm được

Các số trên cùng dòng của input/output file ghi cách nhau bằng dấu trống

Ví dụ:

MAXPATH.INP	MAXPATH.OUT
5 5 1 5	YES
1 2 1	8
2 3 1	
3 4 1	
4 5 1	
2 5 7	

TÌM ĐƯỜNG ĐI ÍT CẠNH NHẤT (LEAST.*)

(Bài tập cơ bản phân đồ thị)

Cho đồ thị có hướng $G=(V,E)$ và hai tập đỉnh $S,T\subset V$. Hãy tìm đường đi qua ít cạnh nhất với đỉnh xuất phát thuộc tập S và đỉnh kết thúc thuộc tập T

Dữ liệu: Vào từ file văn bản LEAST.INP

- Dòng đầu tiên ghi hai số nguyên dương n, m ($n\leq 5\cdot 10^4, m\leq 10^5$) lần lượt là số đỉnh và số cung (một chiều) của đồ thị
- m dòng tiếp theo, mỗi dòng ghi hai số nguyên u, v thể hiện có một cung nối đỉnh u với đỉnh v ($1\leq u, v\leq n$)
- Dòng tiếp theo mô tả các đỉnh của tập S . Đầu tiên là số nguyên k_s - số lượng đỉnh của tập S . Tiếp theo là k_s số nguyên mô tả danh sách các đỉnh của S
- Dòng cuối cùng mô tả tập T : Số đầu tiên k_T là số lượng đỉnh của T , tiếp theo là danh sách các đỉnh trong T

Kết quả: Ghi ra file LEAST.OUT

- Dòng đầu tiên ghi K là số lượng các cạnh trên đường đi tìm được ($K=-1$) nếu như không có đường đi từ tập S đến tập T

Các số trên input/output file cách nhau ít nhất một dấu trống.

Dữ liệu:

LEAST.INP	LEAST.OUT
67	3
12	
23	
13	
34	
45	
56	
46	
11	
16	

Nothing is impossible. Try your best and win.

ĐONG NƯỚC

Nền phẳng của một công trường xây dựng đã được chia thành lưới ô vuông đơn vị kích thước $m \times n$ ô. Trên mỗi ô (i, j) của lưới, người ta dựng một cột bê tông hình hộp có đáy là ô (i, j) và chiều cao là H_{ij} đơn vị. Sau khi dựng xong, thì trời đổ mưa to và đủ lâu. Giả thiết rằng nước không thấm thấu qua các cột bê tông cũng như không rò rỉ qua các đường ghép giữa chúng.

Yêu cầu: Xác định lượng nước đọng giữa các cột

Chú ý kỹ thuật: m, n, H_{ij} là các số nguyên dương. $1 \leq m, n \leq 100$. $1 \leq H_{ij} \leq 1000$

Dữ liệu: Vào từ file văn bản WATER.INP được ghi dưới khuôn dạng sau:

Dòng 1: $m \ n$
Dòng 2: $H_{11} \ H_{12} \ \dots \ H_{1n}$
Dòng 3: $H_{21} \ H_{22} \ \dots \ H_{2n}$
...
Dòng $m + 1$: $H_{m1} \ H_{m2} \ \dots \ H_{mn}$

Các số trên 1 dòng các nhau ít nhất 1 dấu cách

Kết quả: Ghi ra file văn bản WATER.OUT chứa số đơn vị khối nước đọng

Ví dụ:

WATER.INP	WATER.OUT
5 5 9 9 9 9 9 9 2 2 2 9 9 2 1 2 9 9 2 2 2 9 9 9 9 9 9	64

WATER.INP	WATER.OUT
5 7 3 3 3 3 3 3 3 3 1 1 1 1 1 3 3 1 2 2 2 1 3 3 1 1 1 1 1 3 3 3 3 3 3 3 3	27

Bài 3: Sói và Cừu

Có một số con cừu trong trại chăn nuôi của Mickey. Trong khi đang ngủ say, những con sói đã vào trại và tấn công đàn cừu.

Trại có dạng hình chữ nhật gồm các ô tổ chức thành hàng và cột. Ký tự dấu chấm '.' là ô trống, ký tự '#' là hàng rào, ký tự 'o' là cừu và ký tự 'v' là chó sói.

Chúng ta coi hai ô là cùng một miền nếu cả hai ô đó không phải là hàng rào và có thể chuyển từ ô nọ tới ô kia bằng đường đi chỉ gồm các đường đi theo chiều ngang hoặc chiều thẳng đứng không vượt qua hàng rào. Các ô mà từ chúng có thể thoát ra khỏi sân không được xem là một phần của bất kỳ miền nào.

May thay, những con cừu biết tự vệ. Chúng có thể chiến đấu với những con sói trong miền (húc chết sói) nếu số lượng cừu lớn hơn số lượng sói trong cùng miền. Ngược lại những con sói sẽ ăn hết những con cừu trong cùng miền. Với những con cừu có thể thoát ra ngoài biên, chúng vẫn tồn tại và cũng không tấn công lại sói (số lượng sói và cừu không đổi)

Biết vị trí của các con sói và các con cừu. Hãy tính số lượng cừu và sói còn lại trong sáng hôm đó.

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

- Dòng đầu tiên chứa hai số nguyên R và C ($3 \leq R, C \leq 250$) là số hàng và số cột của trại.
- Mỗi dòng trong số R dòng tiếp theo gồm C ký tự. Tất cả các ký tự này biểu diễn các vị trí có hàng rào, cừu, chó sói và vị trí trống trong trại.

Kết quả: Ghi ra file văn bản SOICUU.OUT chỉ một dòng gồm 2 con số: Số cừu và số sói còn lại trong trại.

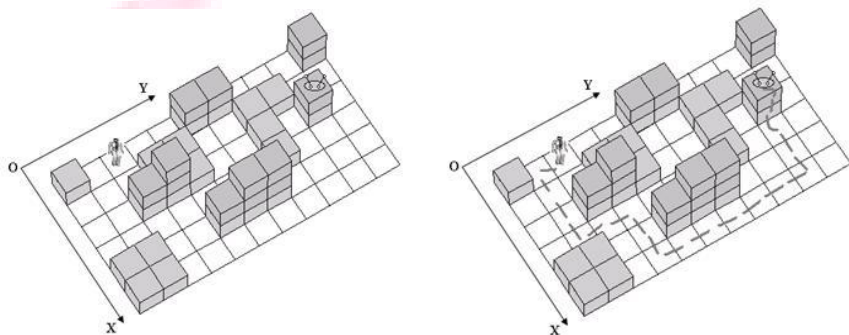
Ví dụ:

SOICUU.INP	SOICUU.INP	SOICUU.INP
6 6 ...#.. .#v#.. #v.#.# #.o#.# .###.# ...##	8 8 .#####. #..o...# #.####.# #.#v.#.# #.#.o#o# #o.##...# #.v..v.# .#####.	9 12 .###.#####.. #.oo#...#v#.. #..o#.#.#.#. #..##o#...#. #.#v#o####.#. #..#v#v#####. .#####.vv.o#####.
SOICUU.OUT	SOICUU.OUT	SOICUU.OUT
0 2	3 1	3 5

Thi Robocom

Trong một cuộc thi Robocon người ta tổ chức phần thi "leo cột" – phần thi thể hiện tốc độ tư duy và kỹ năng, kỹ xảo của các chú Robot. Một sa bàn kích thước $M \times N$ ($0 < M, N < 101$) được chia thành các lưới ô vuông đơn vị, kích thước 1×1 gắn với trục tọa độ OXY như hình vẽ bên dưới. Kích thước M tính theo đơn vị trục OX.

Trên các ô vuông người ta xây các cột hình khối chữ nhật có mặt cắt 1×1 và chiều cao là một số nguyên dương không vượt quá 100, các cột có chiều cao bằng nhau mà giáp nhau thì sẽ tạo thành một mặt bằng.



Khi đến phần thi của mình, mỗi chú Robot được đặt vào một ô vuông tọa độ (x_1, y_1) – đó có thể là bề mặt sa bàn hoặc đỉnh của một cột. nào đó. Tại một vị trí (x_2, y_2) khác người ta đặt một thiết bị phát sóng, các Robot có khả năng bắt sóng và tính ra được vị trí đặt thiết bị này. Từ vị trí (x_1, y_1) hiện tại Robot phải đi đến vị trí (x_2, y_2) , có thể phải leo lên hay leo xuống các cột trên sa bàn và đường mà Robot đi phải là con đường tốn ít năng lượng nhất. Mỗi bước đi của Robot luôn bắt đầu từ tâm một ô vuông này đến tâm một ô vuông khác có chung đúng một cạnh với nó nhưng có thể có 2 khả năng:

- Hai ô vuông thuộc cùng một mặt phẳng: Robot tốn năng lượng là 1.
- Hai ô vuông không thuộc cùng một mặt phẳng – khi đó chúng sẽ vuông góc với nhau: Robot tốn năng lượng là 2 (do phải sử dụng kỹ năng di chuyển phức tạp hơn).

Nhiệm vụ của bạn là phải lập trình tìm ra con đường này. Đối với các chuyên gia Robot việc cài đặt một chương trình như vậy cho Robot không phải khó khăn và bạn có thể hình dung ra trong thực tế những Robot này sẽ có ích như thế nào.

Dữ liệu vào từ file ROBOT.INP:

- Dòng đầu tiên ghi M, N cách nhau bởi một dấu cách.
- M dòng tiếp theo, mỗi dòng ghi N số. Số thứ j trên dòng i thể hiện độ cao của cột có tọa độ (i, j) trên sa bàn. Độ cao 0 thể hiện rằng không có cột nào tại vị trí đó.
- Dòng $M+2$ ghi 4 số x_1, y_1, x_2, y_2 , các số cách nhau bởi một dấu cách

Ghi dữ liệu ra file ROBOT.OUT:

- Ghi một số duy nhất là năng lượng tối thiểu của con đường tìm được cho Robot.

Ví dụ:

Nothing is impossible. Try your best and win.

ROBOT.INP	ROBOT.OUT
6 11 1 0 0 0 0 2 2 0 0 0 2 0 0 0 1 1 0 0 1 1 0 0 0 0 2 3 1 0 0 1 0 2 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 2 3 3 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 3 2 1 0	21

Hai hình vẽ trên thể hiện vị trí ban đầu của Robot và một con đường Robot đi tốn năng lượng 21.



Mê cung

Mê cung có dạng một hình chữ nhật kích thước $m \times n$ được chia ra thành l ô vuông cạnh độ dài 1 bằng các đường song song với các cạnh. Mỗi ô vuông của l ô hoặc là ô cấm hoặc là ô tự do. Từ một ô tự do có thể di chuyển sang các ô tự do có chung cạnh với nó. Không được phép di chuyển vượt khỏi biên của mê cung. Mê cung được thiết kế khá đặc biệt: Giữa hai ô tự do bất kỳ chỉ duy nhất có một cách di chuyển từ ô này đến ô kia. Tại tâm của mỗi ô tự do đều có một cái móc. Trong mê cung có hai ô tự do đặc biệt mà nếu bạn nối được hai cái móc ở hai ô đó bởi một sợi dây thừng thì cánh cửa bí mật của mê cung sẽ tự mở. Vấn đề đặt ra là phải chuẩn bị sợi dây thừng với độ dài ngắn nhất đảm bảo cho dù hai ô đặc biệt có ở vị trí nào trong mê cung bạn vẫn có thể nối được các cái móc ở hai ô đó bằng sợi dây thừng đã chuẩn bị.

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

- Dòng đầu tiên chứa hai số m, n ($3 \leq m, n \leq 1000$)
- Các dòng tiếp theo mô tả mê cung. Dòng thứ i trong số m dòng tiếp theo chứa n ký tự, mỗi ký tự chỉ là '#' hoặc '.', trong đó ký tự '#' cho biết ô ở vị trí tương ứng là ô cấm, còn ký tự '.' cho biết ô ở vị trí tương ứng là ô tự do.

Kết quả: Ghi ra ở trên một dòng của file văn bản LABYR.OUT độ dài (tính bởi số ô) của sợi dây thừng cần chuẩn bị

Ví dụ:

LABYR.INP

#.#
###

LABYR.OUT
0

LABYR.INP
7 6

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

LABYR.OUT
8