

Tìm kiếm theo chiều sâu (DFS).

Cho đơn đồ thị vô hướng liên thông $G = (V, E)$ gồm n đỉnh và m cạnh, các đỉnh được đánh số từ 1 tới n và các cạnh được đánh số từ 1 tới m .

Bằng thuật toán tìm kiếm theo chiều sâu, hãy đưa ra danh sách các đỉnh theo thứ tự tìm kiếm. Biết rằng: Đỉnh nào có chỉ số bé hơn sẽ được ưu tiên thăm trước.

Dữ liệu vào từ file văn bản DFS.INP gồm:

Dòng 1: Ghi 2 số nguyên n, m ($1 \leq n \leq 100$; $1 \leq m \leq 5000$);

M dòng tiếp theo: Mỗi dòng gồm 2 số nguyên i, j mô tả 1 cạnh

Kết quả ghi ra file văn bản DFS.OUT: Gồm N dòng; Mỗi dòng gồm 1 số ghi số hiệu đỉnh theo thứ tự DFS.

Ví dụ:

DFS.INP	DFS.OUT
7 7	1
1 2	2
1 3	4
1 5	6
2 4	5
2 6	3
3 7	7
5 6	

Tìm kiếm theo chiều rộng (BFS)

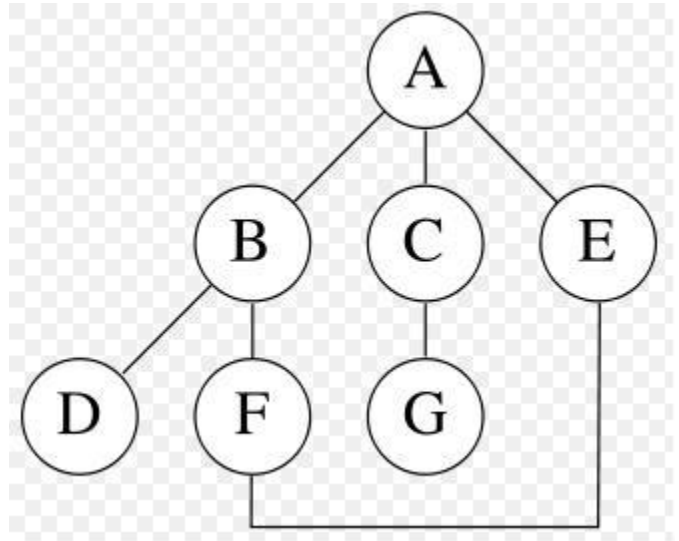
Tìm kiếm ưu tiên chiều rộng hay tìm kiếm theo chiều rộng (tiếng Anh: Breadth-first search - BFS) là một thuật toán duyệt hoặc tìm kiếm trên một cây hoặc một đồ thị.

Thuật toán khởi đầu tại gốc (hoặc chọn một đỉnh nào đó coi như gốc) và phát triển theo nguyên tắc: gần trước xa sau.

Ví dụ:

Tìm kiếm ưu tiên chiều sâu bắt đầu thăm đỉnh A, đi theo cạnh trái, tiếp tục tìm kiếm xong ở cây con trái mới chuyển sang tìm kiếm ở cây con phải. Thứ tự thăm viếng các đỉnh là: A, B, C, E, D, F, G.

Quá trình viếng thăm các đỉnh diễn ra như sau: Sau khi thăm đỉnh A, vì B chưa được thăm nên theo cạnh AB ta thăm B, sau đó ta quay lại A thăm tiếp C, rồi quay lại A thăm tiếp E. Đã hết đỉnh kề với A, ta đi đến B thăm đỉnh D, quay lại B thăm tiếp đỉnh F. Hết đỉnh kề với B ta quay lại B rồi quay lại A, sau đó đến C và thăm tiếp đỉnh G.



Bài toán đặt ra là:

Cho đơn đồ thị vô hướng liên thông $G = (V, E)$ gồm n đỉnh và m cạnh, các đỉnh được đánh số từ 1 tới n và các cạnh được đánh số từ 1 tới m .

Bằng thuật toán tìm kiếm theo chiều rộng, hãy đưa ra danh sách các đỉnh theo thứ tự tìm kiếm.

Biết rằng: Đỉnh nào có chỉ số bé hơn sẽ được ưu tiên thăm trước.

Dữ liệu vào từ file văn bản BFS.INP gồm:

Dòng 1: Ghi 2 số nguyên n, m

M dòng tiếp theo: Mỗi dòng gồm 2 số nguyên i, j mô tả 1 cạnh

Kết quả ghi ra file văn bản BFS.OUT gồm N dòng;

Mỗi dòng gồm 1 số ghi số hiệu đỉnh theo thứ tự BFS.

Ví dụ:

BFS.INP	BFS.OUT
7 7	1
1 2	2
1 3	3
1 5	5
2 4	4
2 6	6
3 7	7
5 6	

Đường đi ngắn nhất.

Cho đơn đồ thị vô hướng liên thông $G = (V, E)$ gồm n đỉnh và m cạnh, các đỉnh được đánh số từ 1 tới n và các cạnh được đánh số từ 1 tới m , và 2 đỉnh s, t

Tìm đường đi ngắn nhất từ s đến t

Dữ liệu vào từ file văn bản MINPATH.INP gồm

Dòng 1: Chứa 4 số n, m, s, t . ($1 \leq n \leq 100$; $1 \leq m \leq 5000$)

M dòng tiếp theo: Dòng thứ i có dạng ba số nguyên u, v, c . Trong đó u, v là chỉ số hai đỉnh đầu mút của cạnh thứ i và c trọng số của cạnh đó ($0 \leq c \leq 10000$).

Kết quả ghi ra file văn bản MINPATH.OUT gồm

Dòng 1: Ghi ra độ dài đường đi ngắn nhất

Dòng 2: Ghi ra thứ tự các đỉnh trên đường đi từ s đến t .

Ví dụ:

MINPATH.INP	MINPATH.OUT
3 3 1 3	4
1 2 3	1 2 3
2 3 1	
1 3 5	

Đua xe Công Thức 1

Thành phố XYZ đang chuẩn bị tổ chức một cuộc đua xe Công Thức 1. Trước khi diễn ra cuộc đua, chính quyền muốn thiết kế một hệ thống gồm một số camera đặt trên các tuyến đường khác nhau để phục vụ truyền hình trực tiếp cho người dân. Để tăng tính hấp dẫn của cuộc đua, cần có ít nhất một camera dọc theo mỗi vòng đua.

Hệ thống đường đua có thể được mô tả bởi một dãy các nút giao thông và các đường nối các nút giao thông là đường hai chiều. Một vòng đua bao gồm một nút giao thông xuất phát, tiếp theo là đường đi bao gồm ít nhất 3 tuyến đường và cuối cùng quay trở lại điểm xuất phát. Trong một vòng đua, mỗi tuyến đường chỉ được đi qua đúng một lần.

Chi phí để đặt camera phụ thuộc vào tuyến đường được chọn, camera được đặt trên các tuyến đường chứ không phải tại các nút giao thông.

Yêu cầu: Cho n nút giao thông, m tuyến đường nối các nút giao thông và chi phí đặt camera trên mỗi tuyến đường. Hãy chọn một số tuyến đường để đặt camera sao cho tổng chi phí lắp đặt là thấp nhất, đồng thời vẫn đảm bảo có ít nhất một camera dọc theo mỗi vòng đua.

Dữ liệu vào từ file 'RACING.INP' có nội dung như sau:

- Dòng đầu tiên chứa hai số nguyên n, m ($1 \leq n \leq 1000$; $1 \leq m \leq 10000$) là số nút giao thông và số tuyến đường. Các nút giao thông được đánh số từ 1 đến n .

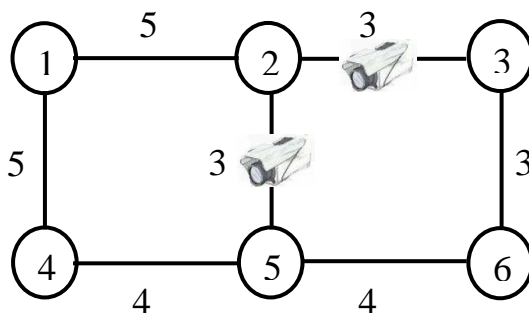
- m dòng tiếp theo mô tả các tuyến đường, mỗi dòng bao gồm 3 số nguyên dương u, v và c , với ý nghĩa là tuyến đường nối hai nút giao thông u và v có chi phí lắp đặt camera là c ($1 \leq c \leq 1000$).

Dữ liệu luôn đảm bảo hai nút giao thông bất kì đều có thể đi đến được với nhau bằng tuyến đường trực tiếp hoặc qua các tuyến đường trung gian. Các số cách nhau bởi một dấu cách.

Kết quả ghi vào file 'RACING.OUT' một số nguyên duy nhất là tổng chi phí lắp đặt thấp nhất tìm được.

Ví dụ:

RACING.INP	RACING.OUT
6 7 1 2 5 2 3 3 1 4 5 4 5 4 5 6 4 6 3 3 5 2 3	6



Rô bốt gỡ mìn.

Trong cuộc chiến tại Afganitan, để hạn chế thương vong cho các binh sĩ, người Mỹ đã áp dụng rất nhiều công nghệ hiện đại trong chiến tranh, một trong số đó là rô bốt gỡ mìn. Trước khi đưa vào sử dụng họ đã thử nghiệm trên một khu đất hình chữ nhật có diện tích $M \times N$ mét vuông. Khu đất được chia thành các ô vuông bằng nhau có diện tích 1 mét vuông, trong đó có 1 ô chứa mìn, một số ô có vật cản và đặt rô bốt vào một ô nào đó của khu đất. Trên màn hình ra đa của rô bốt phát hiện được vị trí có mìn và nhiệm vụ của rô bốt là phải di chuyển thật nhanh đến ô có mìn.

Rô bốt có thể duy chuyển sang các ô chung cạnh hoặc chung góc với ô đang đứng nếu như ô đó không có vật cản. Thời gian đi qua mỗi ô mất 1 giây, hãy tìm đường đi sao cho rô bốt đến được vị trí có mìn sớm nhất có thể.

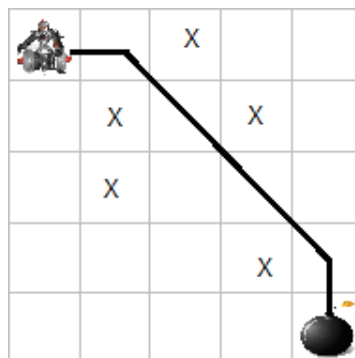
Dữ liệu vào: Từ file văn bản **ROBO.INP** có cấu trúc như sau:

- Dòng 1: chứa 2 số nguyên dương M, N ($2 \leq M, N \leq 1000$).
- Dòng 2: chứa 4 số x_1, y_1, x_2, y_2 ; với x_1, y_1 là tọa độ của ô đặt rô bốt; x_2, y_2 là tọa độ của ô có mìn ($1 \leq x_1, x_2 \leq M; 1 \leq y_1, y_2 \leq N$).
- Các dòng tiếp theo, dòng thứ i chứa 2 số x_i, y_i là tọa độ của ô có vật cản ($1 \leq x_i \leq M; 1 \leq y_i \leq N$).

Dữ liệu ra: Ghi ra file văn bản **ROBO.OUT** một số nguyên duy nhất S là thời gian ít nhất để rô bốt đến được vị trí có mìn. Nếu không có đường đi thì ghi số -1.

Ví dụ:

ROBO.INP	ROBO.INP
5 5	5
1 1 5 5	
1 3	
2 2	
3 2	
2 4	
4 4	



Xây dựng hệ thống lưới điện

Công ty truyền tải điện ABC đang thực hiện dự án cung cấp điện cho một số xã vùng cao của một huyện miền núi. Dự án đã thực hiện được giai đoạn một, trong giai đoạn này công ty đã xây dựng được N trạm biến áp đặt tại N điểm trên địa bàn. Các trạm biến áp này được đánh số thứ tự từ 1 đến N ($2 \leq N \leq 1000$). Hiện nay công ty tiếp tục thực hiện giai đoạn hai là cần phải nối $N-1$ đường dây điện giữa các trạm biến áp này sao cho khi một trạm biến áp bất kỳ được nối với lưới điện quốc gia thì tất cả các trạm đều được cung cấp điện. Hiện tại giữa các trạm biến áp này đã có M ($N \leq M \leq 2000$) con đường bộ đã xây dựng để từ một trạm biến áp này có thể đi đến bất kỳ một trạm biến áp khác và các hộ dân đều đang sống trên những con đường này. Để nối đường dây giữa hai trạm khác nhau, trạm thứ i với trạm thứ j thì có thể cung cấp được điện sinh hoạt cho C_{ij} ($0 \leq C_{ij} \leq 1000$) hộ gia đình sống theo dọc con đường bộ này.

Yêu cầu: Bạn hãy giúp công ty tính toán cách nối dây như thế nào để cung cấp được điện sinh hoạt cho nhiều hộ gia đình nhất và tổng số hộ gia đình được cung cấp điện là bao nhiêu?

Dữ liệu vào: Tập văn bản ELECTRIC.INP gồm:

- + Dòng đầu: ghi số nguyên N và M cách nhau một dấu cách;
- + Dòng thứ k ($1 \leq k \leq M$) trong M dòng còn lại thể hiện thông tin trên con đường thứ k , trong đó mỗi dòng ghi 3 số nguyên cách nhau một dấu cách, số thứ nhất và số thứ 2 là chỉ số của hai trạm được trên con đường đó và số thứ 3 là số hộ dân đang sinh sống trên con đường này.

Dữ liệu ra: Tập văn bản ELECTRIC.OUT

- + Dòng đầu ghi tổng số hộ dân nhiều nhất sẽ được cung cấp điện sinh hoạt;
- + $N-1$ dòng còn lại mỗi dòng ghi một số nguyên là thứ tự của các con đường sẽ được chọn, ghi theo chiều tăng dần từ nhỏ đến lớn.

Ví dụ:

ELECTRIC.INP	ELECTRIC.OUT
3 3	16
1 2 9	1
1 3 6	3
2 3 7	

Hành trình du lịch

Một Công ty du lịch dự định lập một hành trình du lịch xuất phát từ thành phố s với đích đến là thành phố t ($s \neq t$). Để xây dựng hành trình, Công ty sử dụng bản đồ giao thông bao gồm n thành phố (trong đó có hai thành phố s và t) và m tuyến đường hai chiều, mỗi tuyến đường nối hai thành phố. Các thành phố được đánh số từ 1 đến n , các tuyến đường được đánh số từ 1 đến m . Ngoài 2 thành phố s và t , Công ty có một danh sách D gồm p danh thắng nằm ở p trong số n thành phố.

Yêu cầu: Cho biết độ dài của mỗi tuyến đường, hãy giúp Công ty xây dựng hành trình du lịch từ s đến t với tổng độ dài nhỏ nhất, với điều kiện hành trình xây dựng được phải đi qua ít nhất một trong số các thành phố trong danh sách D . Giả thiết là luôn tồn tại đường đi như vậy.

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

- Dòng đầu tiên chứa các số nguyên dương n, m, p, s, t ($2 < n \leq 1000$);
- Dòng thứ i trong số m dòng tiếp theo chứa ba số nguyên u_i, v_i, d_i cho biết tuyến đường thứ i nối hai thành phố u_i, v_i và có độ dài d_i ($0 < d_i \leq 10^5$), $i = 1, 2, \dots, m$;
- Dòng cuối cùng chứa p số nguyên là p chỉ số của các thành phố trong danh sách D . Hai số liên tiếp trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra file văn bản TOUR.OUT một số nguyên là độ dài của hành trình tìm được.

Ví dụ:

TOUR.INP	TOUR.OUT
7 10 3 1 4	18
1 2 1	
1 7 10	
1 5 6	
2 3 2	
3 4 3	
3 7 15	
4 6 7	
4 7 20	
5 6 10	
5 7 11	
5 6 7	

