

Problem statements

The weight matrix



BieuDienMaTranKeDungVecTor_XuatMaTranTrongSo



[Download the code template](#)

Cho một đồ thị có trọng số $G=(V, E)$ với V là tập đỉnh và E là tập cạnh.

Giả sử thông tin đầu vào của bài toán đồ thị gồm số cạnh, số đỉnh, danh sách tên các đỉnh và danh sách các cạnh của đồ thị được nối từ 2 đỉnh có trọng số. **Hãy biểu diễn đồ thị bằng ma trận kề sử dụng cấu trúc vector và xuất ra ma trận trọng số.**

INPUT

- 2 Số nguyên đầu tiên là số cạnh và số đỉnh của đồ thị
- Ở mỗi dòng tiếp theo là 2 chuỗi thể hiện 2 đỉnh của đồ thị được nối với nhau và 1 số nguyên là trọng số của cạnh nối 2 đỉnh.

OUTPUT

- Xuất ra ma trận trọng số

EXAMPLE

Input	Output
7 7	
VN Lao Cam Thai TQ Nhat Han	0 1 2 0 0 0 0
VN Lao 1	1 0 0 0 0 0 0
VN Cam 2	0 0 0 0 0 0 4
Lao VN 1	0 2 0 0 0 0 0
TQ Cam 3	0 0 3 0 0 0 0
Thai Lao 2	0 0 0 0 0 0 0
Cam Han 4	0 0 0 0 0 5 0
Han Nhat 5	
11 9	
VN Lao Cam Thai TQ Indo DTimo Nhat Han	0 1 0 0 0 2 0 0 0
VN Lao 1	0 0 4 0 5 0 0 0 0
VN Indo 2	0 0 0 7 6 0 0 0 0
Han VN 3	0 0 0 0 0 0 9 2 0
Lao Cam 4	0 0 0 0 0 0 0 0 0
Lao TQ 5	0 0 0 0 0 0 1 0 0
Cam TQ 6	0 0 0 0 8 0 0 0 0
Cam Thai 7	0 0 0 0 0 0 0 0 0
DTimo TQ 8	3 0 0 0 0 0 0 0 0
Thai DTimo 9	
Thai Nhat 2	
Indo DTimo 1	

Adjacent vertex

Cho một đồ thị có trọng số $G=(V, E)$ với V là tập đỉnh và E là tập cạnh.

Giả sử thông tin đầu vào của bài toán đồ thị gồm số cạnh, số đỉnh, danh sách tên các đỉnh và danh sách các cạnh của đồ thị được nối từ 2 đỉnh có trọng số.

Hãy biểu diễn đồ thị bằng ma trận kề sử dụng cấu trúc vector và **liệt kê tất cả các đỉnh kề của đỉnh K** (với K là tên 1 đỉnh bất kỳ nhập từ bàn phím)

INPUT

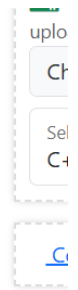
- 2 Số nguyên đầu tiên là số cạnh và số đỉnh của đồ thị
- Ở mỗi dòng tiếp theo là 2 chuỗi thể hiện 2 đỉnh của đồ thị được nối với nhau và 1 số nguyên là trọng số của cạnh nối 2 đỉnh.
- Chuỗi ở dòng cuối cùng là đỉnh cần tìm các đỉnh kề

OUTPUT

- Xuất các đỉnh kề với đỉnh yêu cầu theo mô tả bên dưới
- Nếu không có thì xuất "No find"

EXAMPLE

Input	Output
11 10 Sing Malay Lao Cam Thai TQ Indo DTimo Nhat Han Thai Malay 5 Malay TQ 4 TQ Indo 3 TQ Malay 2 Malay DTimo 1 DTimo TQ 2 DTimo Nhat 3 Malay Cam 4 Cam Nhat 5 Cam Sing 1 Cam Thai 1 VN	No find
10 7 VN Lao Cam Thai Malay Indo DTimo Indo VN 1 Thai Indo 2 VN Thai 3 Malay Indo 2 DTimo Malay 1 Cam DTimo 4 Lao Cam 5 Lao Malay 6 Malay VN 2 VN Lao 1 Lao	Cam Malay



Vertex with degree k

Cho một đồ thị có trọng số $G=(V, E)$ với V là tập đỉnh và E là tập cạnh.

Giả sử thông tin đầu vào của bài toán đồ thị gồm số cạnh, số đỉnh, danh sách tên các đỉnh và danh sách các cạnh của đồ thị được nối từ 2 đỉnh có trọng số.

Hãy biểu diễn đồ thị bằng danh sách kề sử dụng cấu trúc map và **liệt kê tất cả các đỉnh có bậc K** (với K là một giá trị số nguyên bất kỳ nhập từ bàn phím)

INPUT

- 2 Số nguyên đầu tiên là số cạnh và số đỉnh của đồ thị
- Ở mỗi dòng tiếp theo là 2 chuỗi thể hiện 2 đỉnh của đồ thị được nối với nhau và 1 số nguyên là trọng số của cạnh nối 2 đỉnh.
- Số nguyên cuối cùng là bậc K --> cần tìm các đỉnh có bậc K

OUTPUT

- Xuất các đỉnh trong đồ thị có bậc là K
- Nếu không có thì xuất "No find"

EXAMPLE

Input	Output
11 10 Sing Malay Lao Cam Thai TQ Indo DTimo Nhat Han Thai Malay 5 Malay TQ 4 TQ Indo 3 TQ Malay 2 Malay DTimo 1 DTimo TQ 2 DTimo Nhat 3 Malay Cam 4 Cam Nhat 5 Cam Sing 1 Cam Thai 1 0	Lao Han
10 7 VN Lao Cam Thai Malay Indo DTimo Indo VN 1 Thai Indo 2 VN Thai 3 Malay Indo 2 DTimo Malay 1 Cam DTimo 4 Lao Cam 5 Lao Malay 6 Malay VN 2 VN Lao 1 5	No find

DFS traverse from u vertex

Cho một đồ thị có trọng số $G=(V, E)$ với V là tập đỉnh và E là tập cạnh.

Giả sử thông tin đầu vào của bài toán đồ thị gồm số cạnh, số đỉnh, danh sách tên các đỉnh và đỉnh bắt đầu S . **Hãy cài đặt thuật toán duyệt đồ thị theo chiều sâu DFS đi qua tất cả các đỉnh từ đỉnh bắt đầu.**

INPUT

- 2 Số nguyên đầu tiên là số cạnh và số đỉnh của đồ thị
- Ở mỗi dòng tiếp theo là 2 chuỗi thể hiện 2 đỉnh của đồ thị được nối với nhau
- Giá trị cuối cùng là đỉnh bắt đầu duyệt đồ thị

OUTPUT

- Xuất ra kết quả duyệt DFS

EXAMPLE

Input	Output
7 7	
VN Lao Cam Thai TQ Nhat Han	
VN Lao	
VN Cam	
Lao VN	
TQ Cam	TQ Cam Han Nhat
Thai Lao	
Cam Han	
Han Nhat	
TQ	
11 9	
VN Lao Cam Thai TQ Indo DTimo Nhat Han	
VN Lao	
VN Indo	
Han VN	
Lao Cam	Thai Nhat DTimo TQ
Lao TQ	
Cam TQ	
Cam Thai	
DTimo TQ	
Thai DTimo	
Thai Nhat	
Indo DTimo	
Thai	

If there exist path from u to v

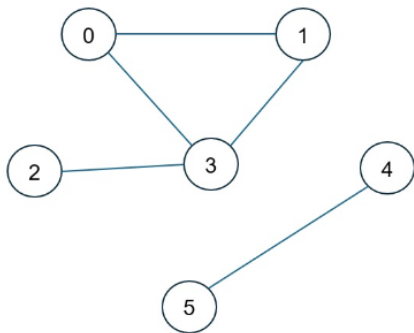
Cho một đồ thị vô hướng G có tập đỉnh $V = \{0, 1, 2, \dots, m\}$ gồm m đỉnh và n cạnh. Hãy viết chương trình cho biết có đường đi từ đỉnh 0 đến các đỉnh còn lại của đồ thị G hay không.

Đầu vào:

- Dòng đầu là hai số m, n cách nhau bằng khoảng trắng. Biết $1 \leq m \leq 1000, 0 \leq n \leq 3000$.
- n dòng kế tiếp, mỗi dòng 2 số nguyên u, v cho biết có cạnh giữa hai đỉnh u và v .

Đầu ra: gồm $m-1$ dòng. Dòng thứ i cho biết giữa đỉnh 0 đến đỉnh i có đường đi hay không. Nếu có thì ghi kết quả "CO", nếu không thì ghi kết quả "KHONG".

Ví dụ: Cho đồ thị như hình bên dưới

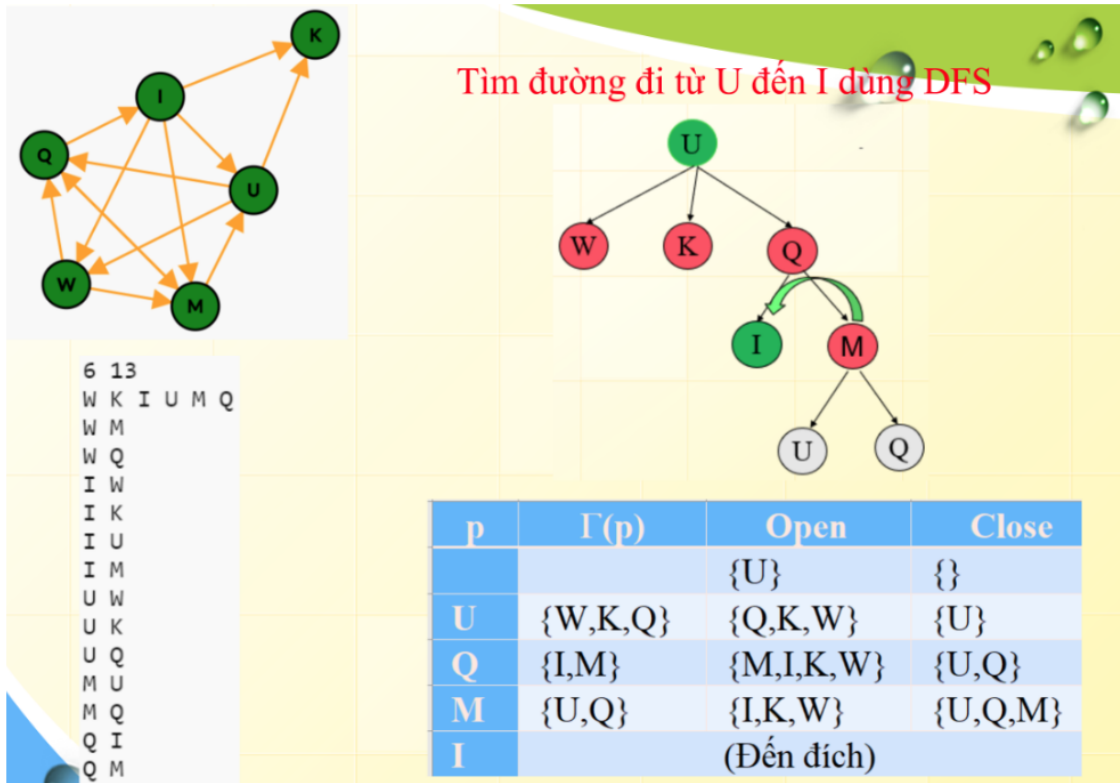


Input	Output
6 5	CO (giải thích: có đường đi từ đỉnh 0 đến đỉnh 1)
0 1	CO (giải thích: có đường đi từ đỉnh 0 đến đỉnh 2)
0 3	CO (giải thích: có đường đi từ đỉnh 0 đến đỉnh 3)
1 3	KHONG (giải thích: không có đường đi từ đỉnh 0 đến đỉnh 4)
2 3	KHONG (giải thích: không có đường đi từ đỉnh 0 đến đỉnh 5)
4 5	

DFS find path from u to v

DFS Tìm đường đi từ u đến v

Các bạn đã tìm hiểu lý thuyết về tìm đường đi trên đồ thị dùng DFS



Viết chương trình nhập vào 1 đồ thị có n đỉnh à e cạnh và m cặp đỉnh đầu, cuối cần tìm đường đi như ví dụ bên dưới
Chương trình xuất ra m chuỗi đường đi dùng DFS, nếu không có đường đi thì ghi ra **no_path**

Sample (Ví dụ):

Input	Output
6 13	U Q I
W K I U M Q	W Q I K
W M	no_path
W Q	
I W	
I K	
I U	
I M	
U W	
U K	
U Q	
M U	
M Q	
Q I	
Q M	
3	
U I	
W K	
K M	

Giải thích đường đi từ U -> I theo DFS

p	F(p)	Open	Close	Previous
	U			
U	W K Q	Q K W	U	U->W, U->K, U->Q
Q	I M	M I K W	U Q	Q->I, Q->M
M	U Q	I K W	U Q M	
I - tìm thấy				

Dựa vào cột Previous thì ta thấy trước I là đỉnh Q, trước đỉnh Q là đỉnh U. Vậy đường đi là U -> Q -> I

Giải thích đường đi từ W -> K theo DFS

p	F(p)	Open	Close	Previous
	W			
W	Q M	M Q	W	W->Q, W->M
M	U Q	U Q	W M	M->U
U	W K Q	K Q	W M U	U->K
K - tìm thấy				

Dựa vào cột Previous thì ta thấy trước K là đỉnh U, trước đỉnh U là đỉnh M, trước đỉnh M là đỉnh W. Do đó đường đi là W -> M -> U -> K

BFS find path from u to v

Viết chương trình nhập vào 1 đồ thị có n đỉnh à e cạnh và m cặp đỉnh đầu, cuối cần tìm đường đi như ví dụ bên dưới
Chương trình xuất ra m chuỗi đường đi dùng BFS, nếu không có đường đi thì ghi ra ***no_path***

Sample (Ví dụ):

Input	Output
6 13	U Q I
W K I U M Q	W M U K
W M	no_path
W Q	
I W	
I K	
I U	
I M	
U W	
U K	
U Q	
M U	
M Q	
Q I	
Q M	
3	
U I	
W K	
K M	

Giải thích đường đi từ U -> I theo BFS

p	F(p)	Open	Close	Previous
		U		
U	W K Q	Q K W	U	U->W, U->K, U->Q
W	M Q	M Q K	U W	W->M
K		M Q	U W K	
Q	I M	I M	U W K Q	Q->I
M	U Q	I	U W K Q M	
I - Tìm thấy				

Dựa vào cột Previous thì ta thấy trước I là đỉnh Q , trước đỉnh Q là đỉnh U. Do đó đường đi là U -> Q -> I

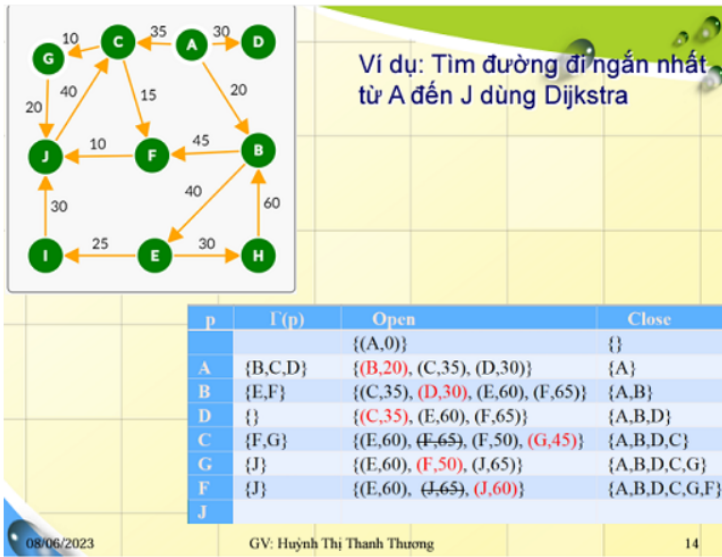
Giải thích đường đi từ W -> K theo BFS

p	F(p)	Open	Close	Previous
		W		
W	M Q	Q M	W	W->M, W->Q
M	U Q	U Q	W M	M->U
Q	I M	I U	W M Q	Q->I
U	W K Q	K I	W M Q U	U->K
I	W K U	K	W M Q U I	
K - Tìm thấy				

Dựa vào cột Previous thì ta thấy trước K là đỉnh U , trước đỉnh U là đỉnh M, trước đỉnh M là đỉnh W. Do đó đường đi là W -> M -> U --> K

Dijkstra

Các bạn đã tìm hiểu lý thuyết về tìm đường đi ngắn nhất trên đồ thị dùng thuật toán Dijkstra



Viết chương trình nhập vào 1 đồ thị có n đỉnh à e cạnh và m cặp đỉnh đầu, cuối cần tìm đường đi như ví dụ bên dưới
Chương trình xuất ra m chuỗi đường đi ngắn nhất dùng Dijkstra và độ dài đường đi ngắn nhất, nếu không có đường đi thì ghi ra **no_path**

Sample (Ví dụ):

Input	Output
10 14	A C F J 60
A B C D E F G H I J	A C F 50
A B 20	
A C 35	
A D 30	
B E 40	
B F 45	
C F 15	
C G 10	
E H 30	
E I 25	
F J 10	
G J 20	
H B 60	
I J 30	
J C 40	
2	
A J	
A F	

Giải thích đường đi từ A -> J

p	F(p)	Open	Close	Previous
		A		
A	B C D	(B,20) (D,30) (C,35)	A	A->B, A->C,A->D
B	E F	(D,30) (C,35) (E,60) (F,65)	A B	A->B, A->C,A->D,B->E, B->F
D		(C,35) (E,60) (F,65)	A B D	
C	F G	(G,45) (E,60) (F,50) (F,65)	A B C D	A->B, A->C,A->D,B->E, C-->F (B->F) ,C-->G
G	J	(F,50) (E,60) (J,65)	A B C D G	
F tìm thấy				

Dựa vào cột Previous thì ta thấy trước F là đỉnh C , trước đỉnh C là đỉnh A. Do đó đường đi là A -> C -> F với độ dài là 50

Hình trong đề có chút nhầm lẫn: sau dòng p(G) là F chứ không phải là J