

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 8 - Thứ tự topo & Ứng dụng](#) / [003. Xếp hạng các đỉnh](#)

Started on	Tuesday, 1 July 2025, 4:58 PM
State	Finished
Completed on	Tuesday, 1 July 2025, 10:10 PM
Time taken	5 hours 12 mins
Marks	1.90/2.00
Grade	9.50 out of 10.00 (95%)

Question 1

Correct

Mark 0.90 out of 1.00

Viết chương trình đọc vào một *đơn đồ thị có hướng không chu trình*, xếp hạng các đỉnh và in hạng của các đỉnh ra màn hình.

Đầu vào (Input)

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên n và m , tương ứng là số đỉnh và số cung.
- m dòng tiếp theo mỗi dòng chứa 2 số nguyên u, v mô tả cung (u, v) .

Đầu ra (Output)

- In ra màn hình hạng của các đỉnh theo thứ tự của đỉnh, mỗi đỉnh trên 1 dòng theo mẫu:

```
r[1] = abc
r[2] = xyz
...
```

Xem thêm ví dụ bên dưới. Trong ví dụ đầu tiên ta có: hạng của 1 = 0, hạng của 2 = 2 và hạng của 3 = 1.

For example:

Input	Result
3 2	r[1] = 0
1 3	r[2] = 2
3 2	r[3] = 1

Answer: (penalty regime: 10, 20, ... %)

```
1 #include <stdio.h>
2 #define MAX_N 100
3
4 //_____LIST_____
5 typedef struct
6 {
7     int data[MAX_N]; // mảng bao gồm các phần tử của danh sách
8     int size;        // độ dài của danh sách
9 } List;
10
11 void make_null_list(List *pL)
12 {
13     pL->size = 0;
14     // (*L).size=0;
15 }
16
17 // Thêm một phần tử mới vào cuối ds
18 void push_back(List *pL, int x)
19 {
20
21     // pL->data[pL->size] = x;
22     // pL->size++;
```

	Input	Expected	Got	
✓	3 2 1 3 3 2	r[1] = 0 r[2] = 2 r[3] = 1	r[1] = 0 r[2] = 2 r[3] = 1	✓
✓	7 10 1 2 1 3 1 4 2 3 2 6 3 7 4 5 5 3 5 7 6 7	r[1] = 0 r[2] = 1 r[3] = 3 r[4] = 1 r[5] = 2 r[6] = 2 r[7] = 4	r[1] = 0 r[2] = 1 r[3] = 3 r[4] = 1 r[5] = 2 r[6] = 2 r[7] = 4	✓

	Input	Expected	Got	
✓	7 12	r[1] = 0	r[1] = 0	✓
	1 2	r[2] = 2	r[2] = 2	
	1 3	r[3] = 1	r[3] = 1	
	2 4	r[4] = 4	r[4] = 4	
	2 5	r[5] = 4	r[5] = 4	
	2 6	r[6] = 3	r[6] = 3	
	3 2	r[7] = 5	r[7] = 5	
	3 5			
	3 6			
	4 7			
	5 7			
	6 4			
	6 5			

Passed all tests! ✓

Question author's solution (C):

```

1  #include <stdio.h>
2
3  #define MAX_N 100
4  typedef struct {
5      int n, m;
6      int A[MAX_N][MAX_N];
7  } Graph;
8
9
10 void init_graph(Graph *pG, int n) {
11     pG->n = n;
12     pG->m = 0;
13     for (int u = 1; u <= n; u++)
14         for (int v = 1; v <= n; v++)
15             pG->A[u][v] = 0;
16 }
17
18 void add_edge(Graph *pG, int u, int v) {
19     pG->A[u][v] += 1;
20 }
21
22

```

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.90/1.00**.

Question 2

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc vào một *đa đồ thị có hướng không chu trình*, xếp hạng các đỉnh và in hạng của các đỉnh ra màn hình.

Đầu vào (Input)

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên n và m, tương ứng là số đỉnh và số cung.
- m dòng tiếp theo mỗi dòng chứa 2 số nguyên u, v mô tả cung (u, v).

Đầu ra (Output)

- In ra màn hình hạng của các đỉnh theo thứ tự của đỉnh, mỗi đỉnh trên 1 dòng theo mẫu:

```
r[1] = abc
r[2] = xyz
...
```

Xem thêm ví dụ bên dưới. Trong ví dụ đầu tiên ta có: hạng của 1 = 0, hạng của 2 = 2 và hạng của 3 = 1.

Gợi ý

- Xem tài liệu thực hành.

For example:

Input	Result
3 4	r[1] = 0
1 3	r[2] = 2
3 2	r[3] = 1
1 3	
1 3	

Answer: (penalty regime: 10, 20, ... %)

```
1 #include <stdio.h>
2 #define MAX_N 100
3
4 // _____ LIST _____
5 typedef struct
6 {
7     int data[MAX_N]; // mảng bao gồm các phần tử của danh sách
8     int size;        // độ dài của danh sách
9 } List;
10
11 void make_null_list(List *pL)
12 {
13     pL->size = 0;
14     // (*L).size=0;
15 }
16
17 // Thêm một phần tử mới vào cuối ds
18 void push_back(List *pL, int x)
19 {
20
21     // pL->data[pL->size] = x;
22     // pL->size++;
```

	Input	Expected	Got	
✓	3 4	r[1] = 0	r[1] = 0	✓
	1 3	r[2] = 2	r[2] = 2	
	3 2	r[3] = 1	r[3] = 1	
	1 3			
	1 3			

	Input	Expected	Got	
✓	7 12 1 2 1 3 1 4 2 3 2 6 3 7 4 5 5 3 5 7 6 7 3 7 1 4	r[1] = 0 r[2] = 1 r[3] = 3 r[4] = 1 r[5] = 2 r[6] = 2 r[7] = 4	r[1] = 0 r[2] = 1 r[3] = 3 r[4] = 1 r[5] = 2 r[6] = 2 r[7] = 4	✓
✓	7 15 1 2 1 3 2 4 2 5 2 6 3 2 3 5 3 6 4 7 5 7 6 4 6 5 6 4 2 4 2 4	r[1] = 0 r[2] = 2 r[3] = 1 r[4] = 4 r[5] = 4 r[6] = 3 r[7] = 5	r[1] = 0 r[2] = 2 r[3] = 1 r[4] = 4 r[5] = 4 r[6] = 3 r[7] = 5	✓

Passed all tests! ✓

Question author's solution (C):

```

1 #include <stdio.h>
2
3 #define MAX_N 100
4 typedef struct {
5     int n, m;
6     int A[MAX_N][MAX_N];
7 } Graph;
8
9
10 void init_graph(Graph *pG, int n) {
11     pG->n = n;
12     pG->m = 0;
13     for (int u = 1; u <= n; u++)
14         for (int v = 1; v <= n; v++)
15             pG->A[u][v] = 0;
16 }
17
18 void add_edge(Graph *pG, int u, int v) {
19     pG->A[u][v] += 1;
20 }
21
22

```

Correct

Marks for this submission: 1.00/1.00.

◀ 002. Xếp đá

Jump to...

004. Ứng dụng xếp hạng ▶