

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 8 - Thứ tự topo & Ứng dụng](#) / [005. Quản lý dự án](#)

<b>Started on</b>	Tuesday, 1 July 2025, 10:37 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 1 July 2025, 10:38 PM
<b>Time taken</b>	8 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>10.00</b> out of 10.00 ( <b>100%</b> )

## Question 1

Correct

Mark 1.00 out of 1.00

Cho một dự án gồm  $n$  công việc. Mỗi công việc  $u$  có một thời gian hoàn thành  $d[u]$  và một danh sách các công việc phải hoàn thành trước khi thực hiện  $u$ . Hãy tính thời gian sớm nhất và thời gian trễ nhất để bắt đầu các công việc.

**Đầu vào (Input)**

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

- Dòng đầu tiên chứa 1 số nguyên  $n$  là số công việc.
- $n$  dòng tiếp theo, dòng thứ  $u$  chứa số nguyên  $d[u]$  là thời gian cần thiết để hoàn thành công việc  $u$  và một danh sách các công việc cần phải hoàn thành trước khi bắt đầu công việc  $u$ . Danh sách được kết thúc bằng số 0.

**Đầu ra (Output)**

- In ra màn hình  $n$  dòng, dòng thứ  $u$  gồm 2 số nguyên  $t[u]$  (thời gian sớm nhất để bắt đầu công việc  $u$ ) và  $T[u]$  (thời gian trễ nhất để bắt đầu công việc  $u$ ).

For example:

Input	Result
10	0 0
7 0	7 11
3 1 0	10 14
1 2 0	7 7
8 1 0	15 18
2 3 4 0	15 15
1 3 4 0	15 19
1 3 4 0	16 16
2 6 0	18 18
2 8 0	20 20
1 5 7 9 0	

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

```

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

```

	Input	Expected	Got	
✓	10	0 0	0 0	✓
	7 0	7 11	7 11	
	3 1 0	10 14	10 14	
	1 2 0	7 7	7 7	
	8 1 0	15 18	15 18	
	2 3 4 0	15 15	15 15	
	1 3 4 0	15 19	15 19	
	1 3 4 0	16 16	16 16	
	2 6 0	18 18	18 18	
	2 8 0	20 20	20 20	
	1 5 7 9 0			

	Input	Expected	Got	
✓	7 8 0 17 4 1 0 15 6 2 7 5 4 0 14 0 7 0 18 4 5 0 1 6 2 0	0 7 14 15 33 33 0 0 0 7 14 14 32 32	0 7 14 15 33 33 0 0 0 7 14 14 32 32	✓
✓	7 8 5 0 19 6 5 0 19 7 2 6 0 17 7 5 2 3 0 10 0 3 0 20 1 6 0	10 10 10 19 38 38 57 57 0 0 0 15 18 18	10 10 10 19 38 38 57 57 0 0 0 15 18 18	✓

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.

◀ 004. Ứng dụng xếp hạng

Jump to...

006. Quản lý dự án phần mềm ▶