

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 8 - Thứ tự topo & Ứng dụng](#) / [006. Quản lý dự án phần mềm](#)

Started on	Tuesday, 1 July 2025, 10:38 PM
State	Finished
Completed on	Tuesday, 1 July 2025, 10:39 PM
Time taken	12 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

Khôi là người quản lý dự án của công ty phần mềm Grafity Ltd. Nhiệm vụ của anh là nhận dự án, phân tích và chia phân công các lập trình viên thực hiện nó. Khi nhận được dự án, việc đầu tiên của Khôi là phân chia dự án thành các công việc và ước lượng thời gian hoàn thành từng công việc. Ngoài ra, tùy theo dự án, các công việc này có thể phụ thuộc nhau ví dụ: công việc A cần phải làm xong rồi thì mới có thể thực hiện công việc B.

Sau khi có được danh sách các công việc, thời gian hoàn thành các công việc và sự phụ thuộc giữa chúng. Khôi sẽ ước lượng được cần ít nhất bao nhiêu thời gian để hoàn thành dự án. Làm thủ công khá mất thời gian nên Khôi mới nhờ đến bạn. Hãy lập trình để giúp anh ấy.

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.
- Dòng thứ 2 chứa n số nguyên tương ứng với thời gian hoàn thành của n công việc
- Dòng thứ 3 chứa 1 số nguyên m cho biết số cặp công việc phụ thuộc nhau
- m dòng tiếp theo, mỗi dòng chứa 2 số nguyên A B nói rằng phải hoàn thành công việc A xong thì mới có thể bắt đầu B..

Đầu ra (Output)

- In ra màn hình một số nguyên duy nhất cho biết tổng thời gian ít nhất để hoàn thành dự án.

For example:

Input	Result
10 7 3 1 8 2 1 1 2 2 1 14 1 2 1 4 2 3 4 5 4 6 4 7 3 5 3 6 3 7 6 8 8 9 9 10 6 10 7 10	21

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 7 3 1 8 2 1 1 2 2 1 14 1 2 1 4 2 3 4 5 4 6 4 7 3 5 3 6 3 7 6 8 8 9 9 10 6 10 7 10	21	21	✓
✓	7 8 6 8 6 20 10 13 10 5 1 3 2 4 2 1 2 4 3 1 3 6 7 3 7 4 7 5 7	49	49	✓

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.

◀ 005. Quản lý dự án

Jump to...

Cần đá ▶