

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 8 - Thứ tự topo & Ứng dụng](#) / [Tổ chức thi công - Dự án xây nhà](#)

<b>Started on</b>	Tuesday, 1 July 2025, 10:41 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 1 July 2025, 10:41 PM
<b>Time taken</b>	4 secs
<b>Marks</b>	0.00/1.00
<b>Grade</b>	<b>0.00</b> out of 10.00 (0%)

## Question 1

Not answered

Mark 0.00 out of 1.00

Có một dự án xây nhà với 10 công việc được cho như bảng sau:

Công việc	Nội dung công việc	Thời gian thực hiện d(i) tính theo tuần	Công việc trước đó
A	Các công việc hồ (nề)	7	Ø
B	Dựng khung cho mái	3	A
C	Lợp mái	1	B
D	Lắp đặt hệ thống vệ sinh, chiếu sáng	8	A
E	Trang trí mặt tiền	2	C, D
F	Ráp cửa sổ	1	C, D
G	Trang hoàng vườn	1	C, D
H	Làm trần	2	F
J	Sơn phết	2	H
K	Chuyển nhà	1	E, G, J

**Người ta cần:**

- Xác định thời điểm sớm nhất và trễ nhất để bắt đầu cho mỗi công việc mà không ảnh hưởng đến tiến độ của dự án  
Để đơn giản trong cài đặt, ta đánh số lại các công việc theo thứ tự 1, 2, 3 ...thay vì A, B, C.....như sau:

A	B	C	D	E	F	G	H	J	K	$\alpha$	$\beta$
1	2	3	4	5	6	7	8	9	10	11	12

và lưu vào tập tin theo định dạng như bảng giá trị **đầu vào**

Hãy viết chương trình tìm **thời gian sớm nhất hoàn thành dự án** và **Thời điểm sớm nhất và trễ nhất để bắt đầu cho mỗi công việc của dự án** mà không ảnh hưởng đến tiến độ của dự án.

Đầu vào:

```
10
7 0
3 1 0
1 2 0
8 1 0
2 3 4 0
1 3 4 0
1 3 4 0
2 6 0
2 8 0
1 5 7 9 0
```

Dòng đầu tiên là số công việc (10), các dòng tiếp theo mỗi dòng mô tả một công việc bao gồm d[u]: thời gian hoàn thành công việc u và danh sách các công việc trước đó của u. Danh sách được kết thúc bằng số 0. Ví dụ: công việc 1 (công việc A) có d[1] = 7 và danh sách các công việc trước đó rỗng.

Công việc 2 (công việc B) có d[2] = 3 và danh sách công việc trước đó là {1}.

Đầu ra:

Dòng đầu tiên: Thời gian sớm nhất hoàn thành dự án

Mỗi dòng tiếp theo: In ra thời gian sớm nhất và thời trễ nhất để bắt đầu cho mỗi công việc (1 => n+2, gồm cả công việc alpha và beta)

**t(u)-T(u)**

Chú ý đọc dữ liệu:

```

freopen("DuAnXayNha.txt", "r", stdin);
scanf("%d", &n);
init_graph(&G, n+2);
for(u=1; u<=n; u++){
    scanf("%d", &d[u]);
    do{
        scanf("%d", &v);
        if(v>0)
            add_edge(&G, v, u);
    }while(v>0);
}

```

For example:

Input	Result
10	21
7 0	0-0
3 1 0	7-11
1 2 0	10-14
8 1 0	7-7
2 3 4 0	15-18
1 3 4 0	15-15
1 3 4 0	15-19
2 6 0	16-16
2 8 0	18-18
1 5 7 9 0	20-20
	0-0
	21-21

**Answer:** (penalty regime: 33.3, 66.7, ... %)

1		
---	--	--

◀ Cân đá

Jump to...

Tổ chức thi công - Dự án phần mềm ▶