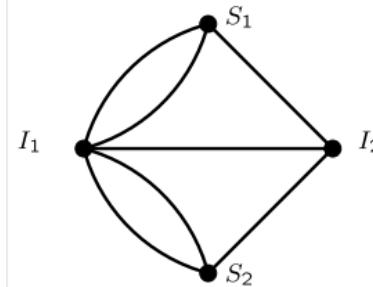
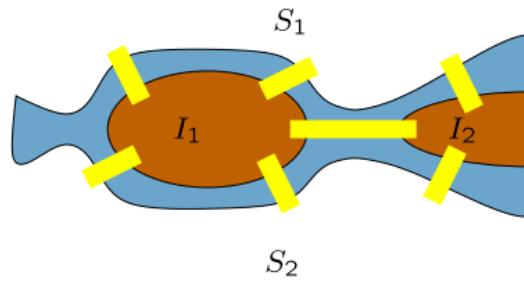


LÝ THUYẾT ĐỒ THỊ

Thứ tự topo & Ứng dụng

Phạm Nguyên Khang
BM. Khoa học máy tính, CNTT
pnkhang@cit.ctu.edu.vn



Cần Thơ, 8/2021

Nội dung

- Lập kế hoạch học tập
- Thứ tự topo
- Xếp hạng kết quả thi đấu
- Xếp hạng các đỉnh của đồ thị
- Quản lý dự án

Lập kế hoạch học tập

Mã số học phần	Tên học phần	Học phần tiên quyết
CT101	Lập trình căn bản A	
CT172	Toán rời rạc	
CT103	Cấu trúc dữ liệu	CT101
CT174	Phân tích và thiết kế thuật toán	CT103
CT175	Lý thuyết đồ thị	CT103
CT176	Lập trình hướng đối tượng	CT101
CT171	Nhập môn công nghệ phần mềm	
CT173	Kiến trúc máy tính	
CT178	Nguyên lý hệ điều hành	CT173
CT112	Mạng máy tính	CT178
CT179	Quản trị hệ thống	
CT180	Cơ sở dữ liệu	CT103
CT109	Phân tích và thiết kế hệ thống thông tin	CT180

Lập kế hoạch học tập

Mã số học phần	Tên học phần	Học phần tiên quyết
CT101	Lập trình căn bản A	
CT172	Toán rời rạc	
CT103	Cấu trúc dữ liệu	
CT174	Phân tích và thiết kế hệ thống	
CT175	Lý thuyết đồ thị	
CT176	Lập trình hướng đối tượng	
CT171	Nhập môn cơ khí	
CT173	Kiến trúc máy tính	
CT178	Nguyên lý điện lạnh	CT173
CT112	Mạng máy tính	CT178
CT179	Quản trị hệ thống	
CT180	Cơ sở dữ liệu	CT103
CT109	Phân tích và thiết kế hệ thống thông tin	CT180



**Lập kế hoạch
học tập thoả
mãn điều kiện
tiên quyết**

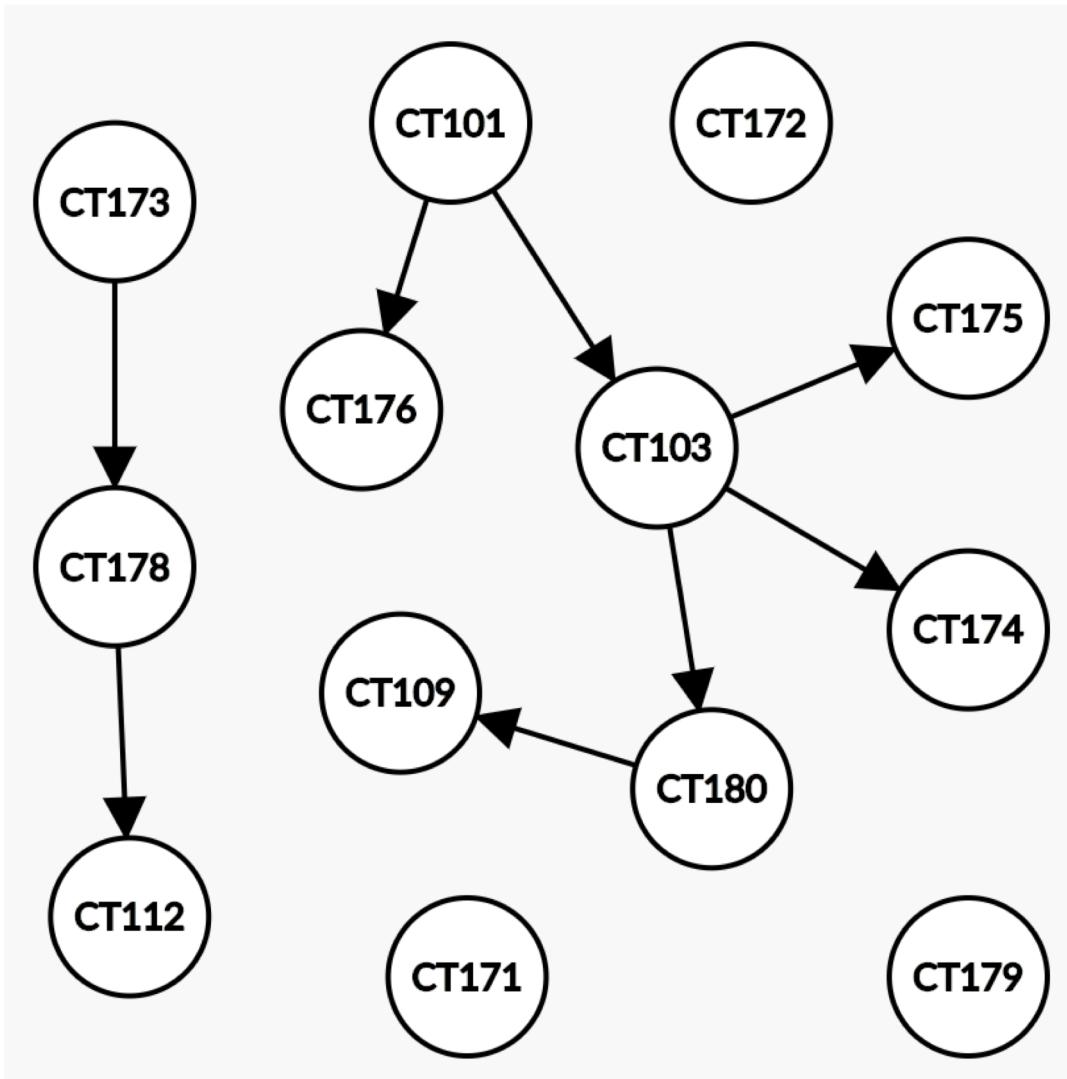
Lập kế hoạch học tập

Mã số học phần	Tên học phần	Học phần tiên quyết
CT101	Lập trình căn bản A	
CT172	Toán rời rạc	
CT103	Cấu trúc dữ liệu	
CT174	Phân tích và thiết kế	
CT175	Lý thuyết đồ thị	
CT176	Lập trình hướng đối tượng	
CT171	Nhập môn cơ khí	
CT173	Kiến trúc máy tính	
CT178	Nguyên lý hệ điều hành	
CT112	Mạng máy tính	

**Lập kế hoạch
học tập thoả
mãn điều kiện
tiên quyết**

Sắp xếp các học phần sao cho các HP tiên quyết phải
đứng trước HP phụ thuộc

Lập kế hoạch học tập



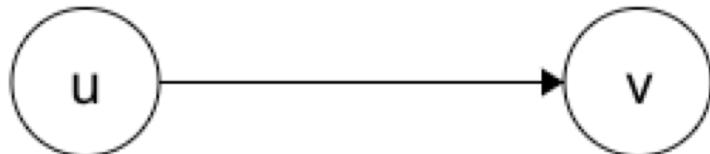
Học phần => Đỉnh
QH tiên quyết => Cung

**Đồ thị mô tả mối quan hệ
tiên quyết giữa các học
phân**

Cung của ĐT có hướng (nhắc lại)

Đỉnh đầu (start/initial vertex)

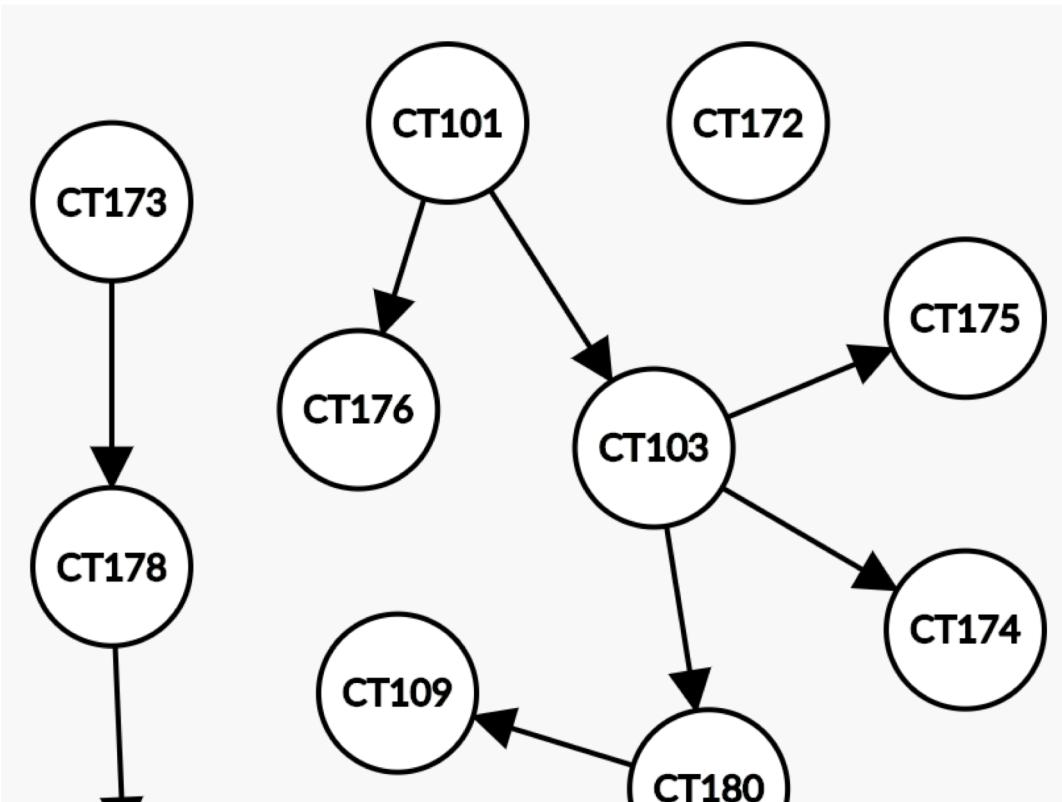
Đỉnh gốc



Đỉnh cuối (end/terminal vertex)

Đỉnh ngọn

Lập kế hoạch học tập

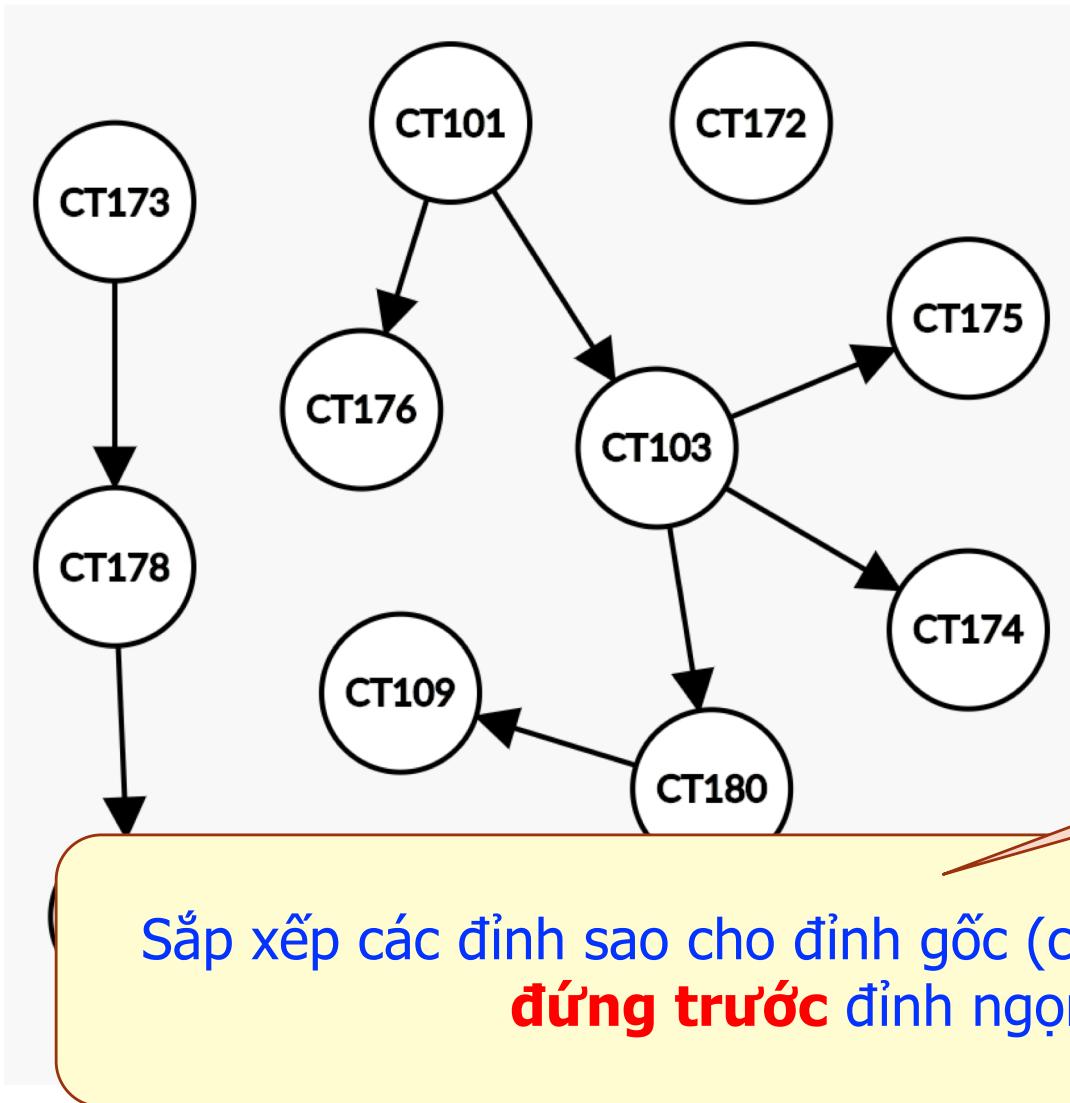


Học phần => Đỉnh
QH tiên quyết => Cung

**Đồ thị mô tả mối quan hệ
tiên quyết giữa các học
phân**

Sắp xếp các đỉnh sao cho đỉnh gốc (của 1 cung) phải
đứng trước đỉnh ngọn

Lập kế hoạch học tập



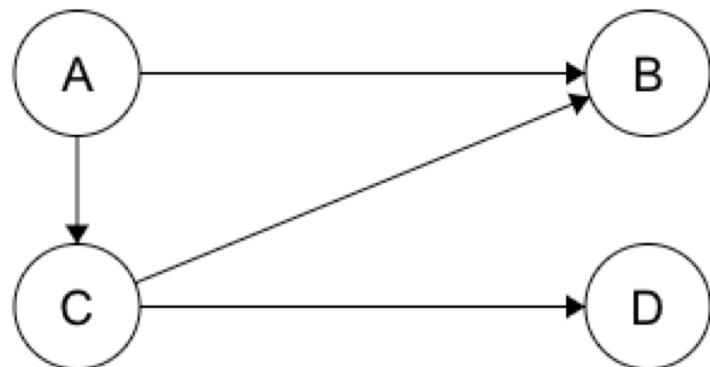
Học phần => Đỉnh
QH tiên quyết => Cung

Đồ thị mô tả mối quan hệ
tiên quyết giữa các học
phân

Thứ tự topo

Thứ tự topo

- **Thứ tự topo** (topological order) của các đỉnh trên đồ thị có hướng G là một cách xếp thứ tự các đỉnh của G sao cho với mọi cung (u, v) , đỉnh gốc u phải đứng trước đỉnh ngọn v.
- **Sắp xếp topo** (topological ordering/sort): sắp xếp các đỉnh theo thứ tự topo

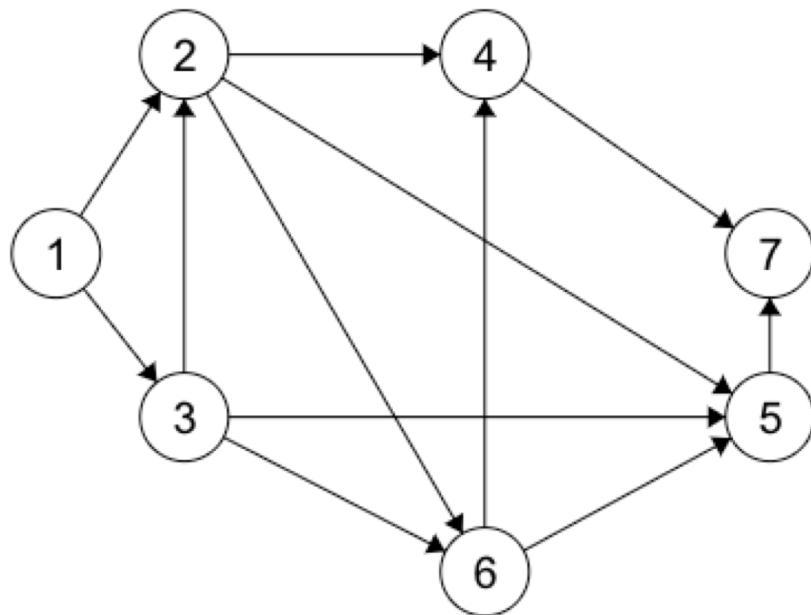


Thứ tự topo:

- A, C, B, D hoặc
- A, C, D, B

Thứ tự topo

- Tìm 1 thứ tự topo của đồ thị bên



Thứ tự topo

- Đồ thị có hướng không chu trình (Directed Acyclic Graph - DAG)
 - Có hướng
 - Không chứa chu trình
 - Tồn tại ít nhất 1 thứ tự topo

Thứ tự topo

- Thuật toán sắp xếp topo dựa trên BFS (Kahn, 1962)
- **Khởi tạo**
Tính bậc vào của các đỉnh, $d[u]$
 Q chứa các đỉnh có bậc vào = 0
 L = rỗng
- **Lặp**
while (Q khác rỗng)
Lấy phần tử đầu hàng đợi Q , gọi nó là u
Thêm u vào L
for các đỉnh kề v của u
Giảm bậc vào của đỉnh v
if bậc vào của v = 0
Thêm v vào Q

Thứ tự topo

- Thuật toán sắp xếp topo dựa trên DFS

DFS(u)

mark[u] = 1 //1. Đánh dấu đã duyệt

for các đỉnh kề v của u //2. Xét các đỉnh kề

if mark[v] == 0

DFS(v)

Đưa u vào L

//3. Sau khi duyệt xong,
đưa u vào L

Đảo ngược danh sách L => Thứ tự topo

Xếp hạng kết quả thi đấu

- Xét một trò chơi đối kháng (ví dụ: cờ vua, bi-a) có n người tham dự
 - Bốc thăm thành m cặp thi đấu với nhau, kết quả:
 - A thắng B (B thua A)
 - B thắng C (C thua B)
 - ...
- Xếp hạng
 - Hạng nhất
 - Hạng nhì
 - ...



A thắng B => hạng A phải cao hơn B!

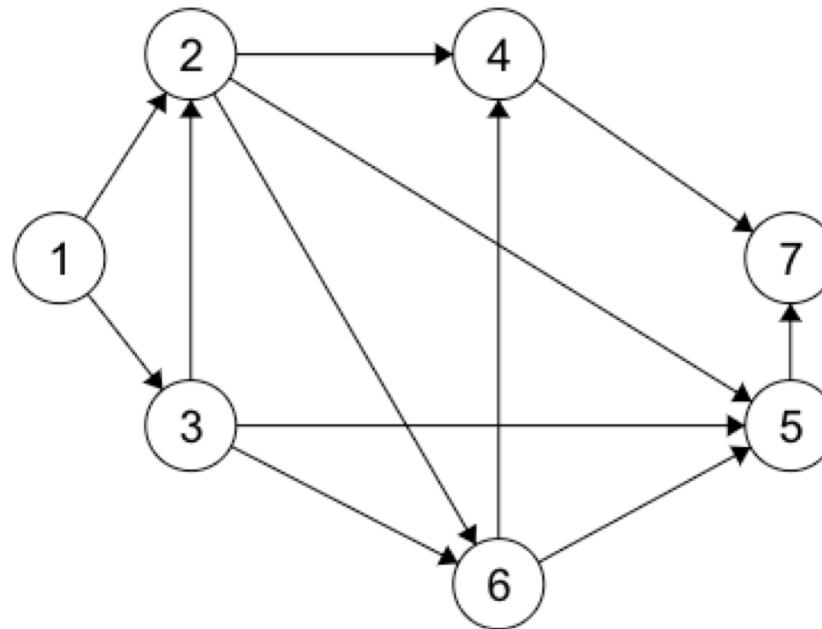
Xếp hạng kết quả thi đấu

- Hạng nhất
 - Không thua ai cả
- Hạng nhì
 - Chỉ thua hạng nhất
- Hạng ba
 - Chỉ thua hạng nhì và hạng nhất (nếu có)
- ...
- Hạng k
 - Chỉ thua hạng k – 1 và các hạng trước đó (nếu có)



Kết quả thi đấu => Đồ thị

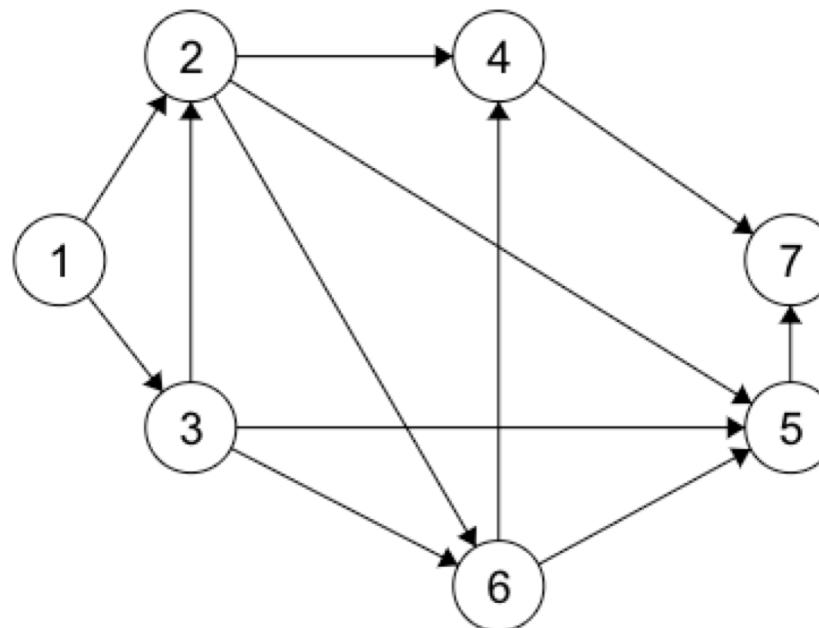
- Người tham dự ~ Đỉnh
- Kết quả của các trận đấu ~ Cung
 - Ví dụ: A thắng B ~ cung (A, B)



Xếp hạng người tham dự => Xếp hạng các đỉnh

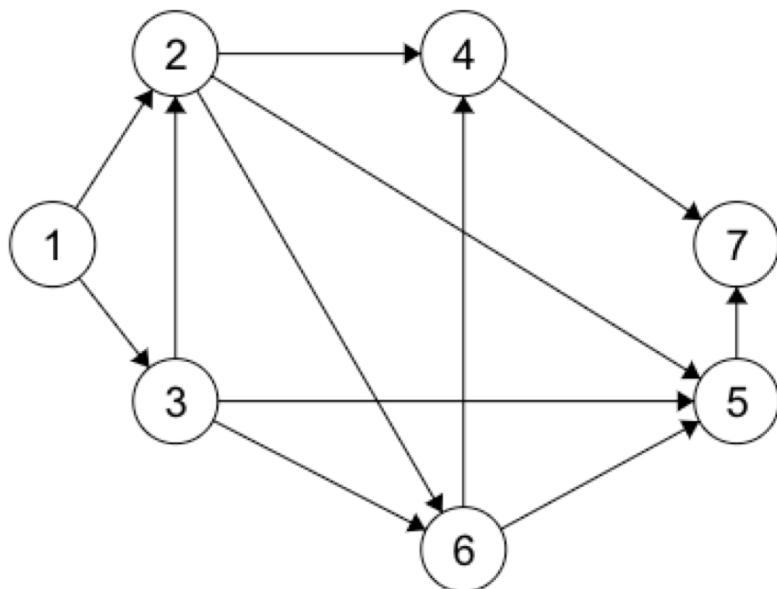
Xếp hạng các đỉnh của đồ thị

- Gốc/nguồn (root/source) của đồ thị
 - Đỉnh không có cung đi vào (bậc vào = 0)
 - DAG có ít nhất 1 gốc



Xếp hạng các đỉnh của đồ thị

- Hạng (rank) của 1 đỉnh
 - u là gốc, $\text{rank}[u] = 0$
 - ngược lại, $\text{rank}[u] = \text{số cung trên} \text{ } \textcolor{red}{\textit{đường đi dài nhất}} \text{ từ gốc đến } u$

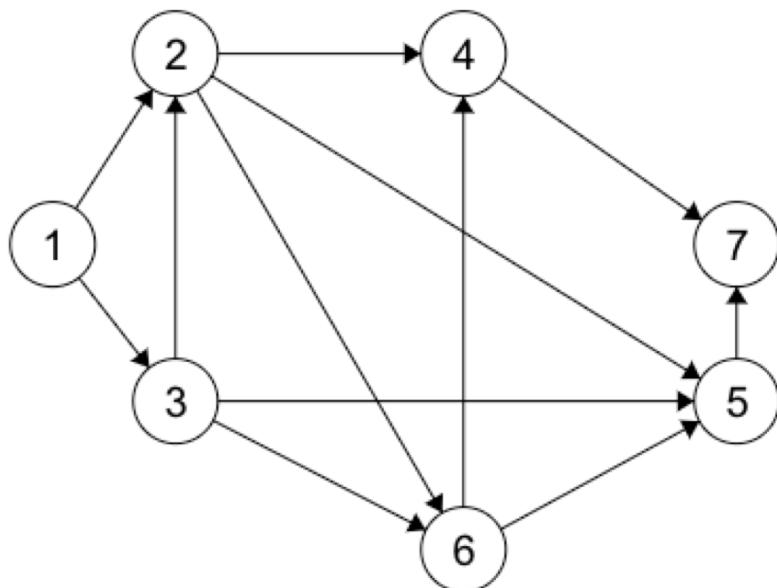


(Các) đỉnh nào là gốc của đồ thị ?

Tìm hạng của các đỉnh

Xếp hạng các đỉnh của đồ thị

- Hạng (rank) của 1 đỉnh
 - u là gốc, $\text{rank}[u] = 0$
 - ngược lại, $\text{rank}[u] = \text{số cung trên đường đi dài nhất từ 1 gốc đến } u$



(Các) đỉnh nào là gốc của đồ thị ?

1

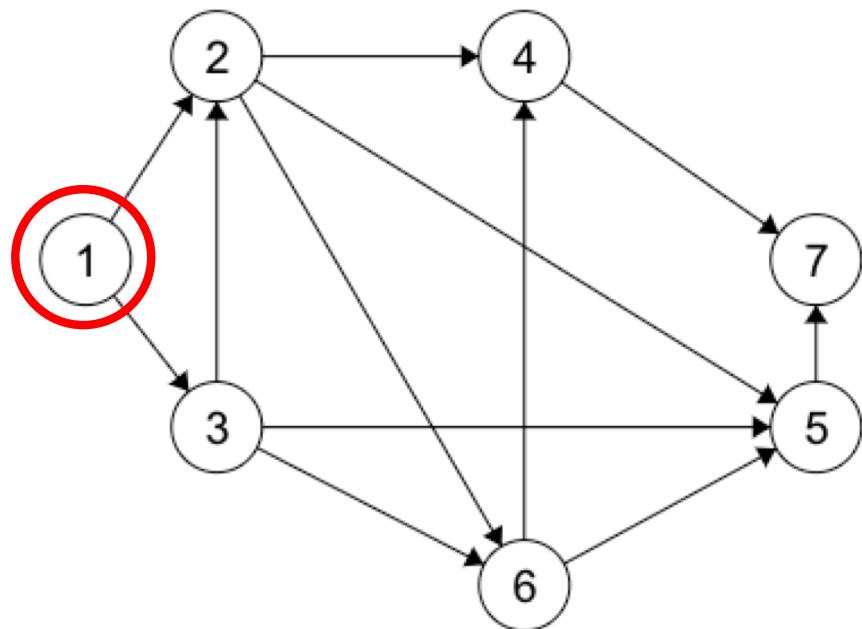
Tìm hạng của các đỉnh

$\text{rank}[1] = 0$
 $\text{rank}[2] = 2$
 $\text{rank}[3] = 1$
 $\text{rank}[4] = 4$
 $\text{rank}[5] = 4$
 $\text{rank}[6] = 3$
 $\text{rank}[7] = 5$

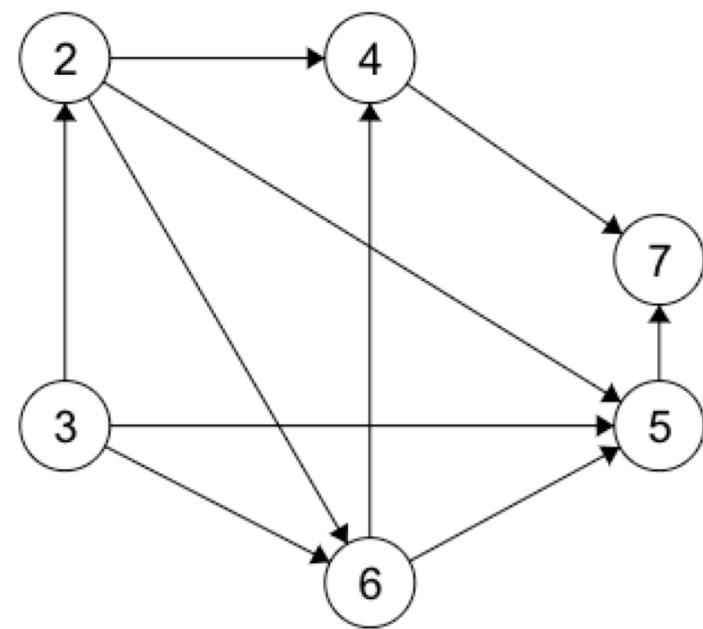
Xếp hạng các đỉnh của đồ thị

- Thuật toán xếp hạng đồ thị (dựa trên BFS)
 - Xếp hạng cho (các) gốc (hạng 0)
 - Loại bỏ gốc ra khỏi đồ thị => xuất hiện gốc mới
 - Xếp hạng cho gốc mới (hạng 1)
 - Loại bỏ gốc ra khỏi đồ thị => xuất hiện gốc mới
 - ...

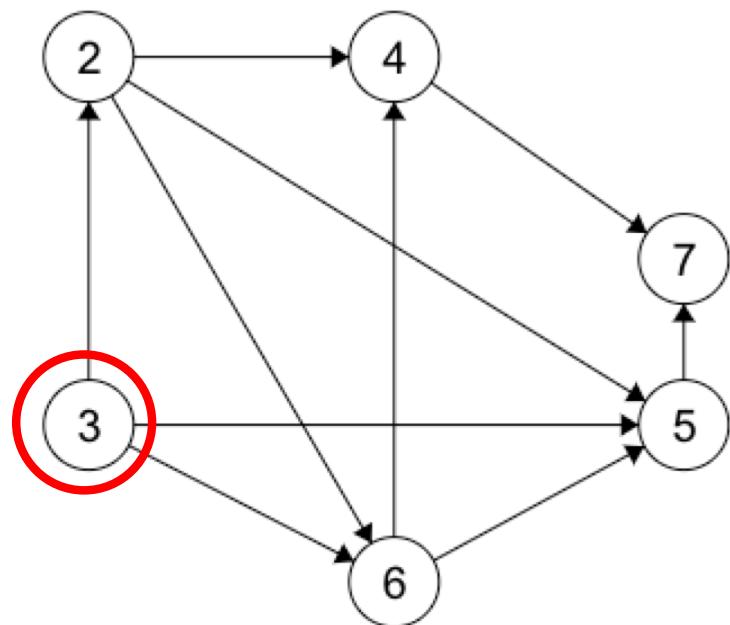
Xếp hạng các đỉnh của đồ thị



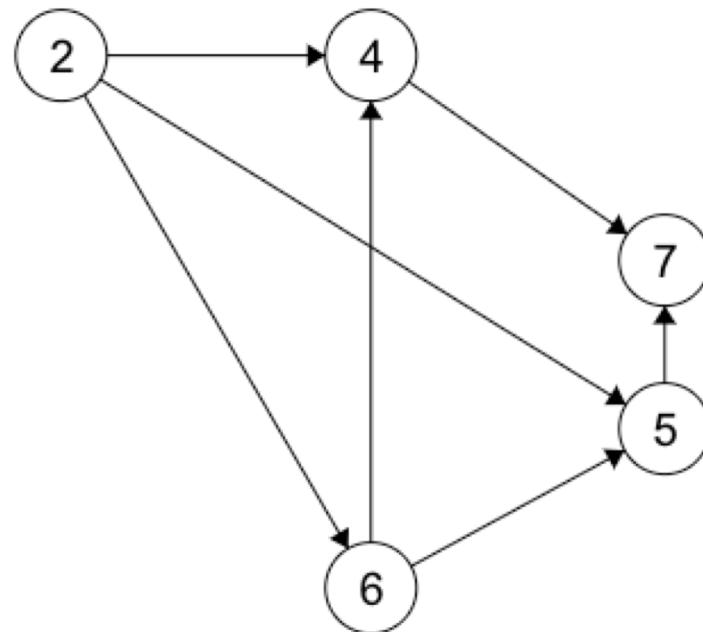
rank[1] = 0



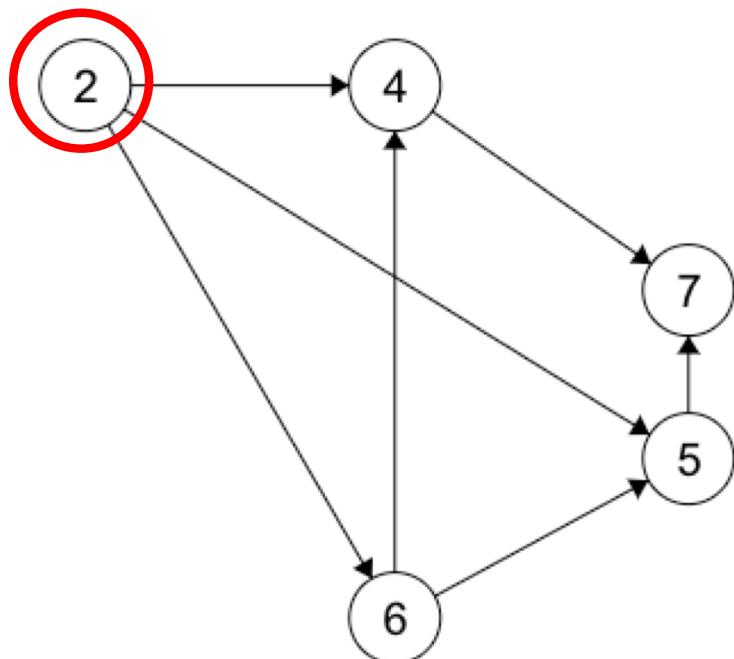
Xếp hạng các đỉnh của đồ thị



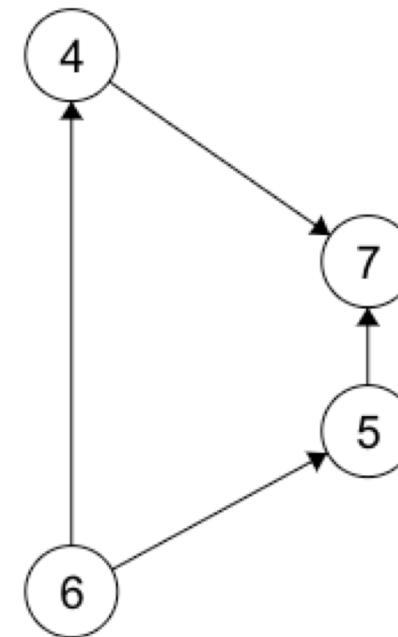
rank[3] = 1



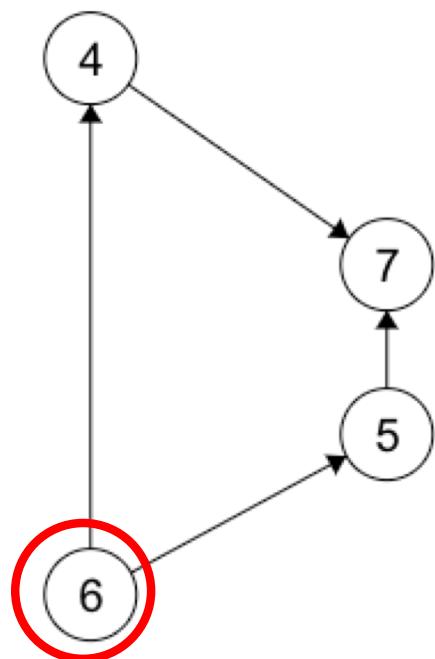
Xếp hạng các đỉnh của đồ thị



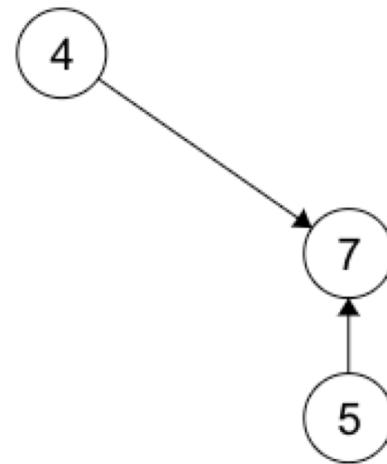
rank[2] = 2



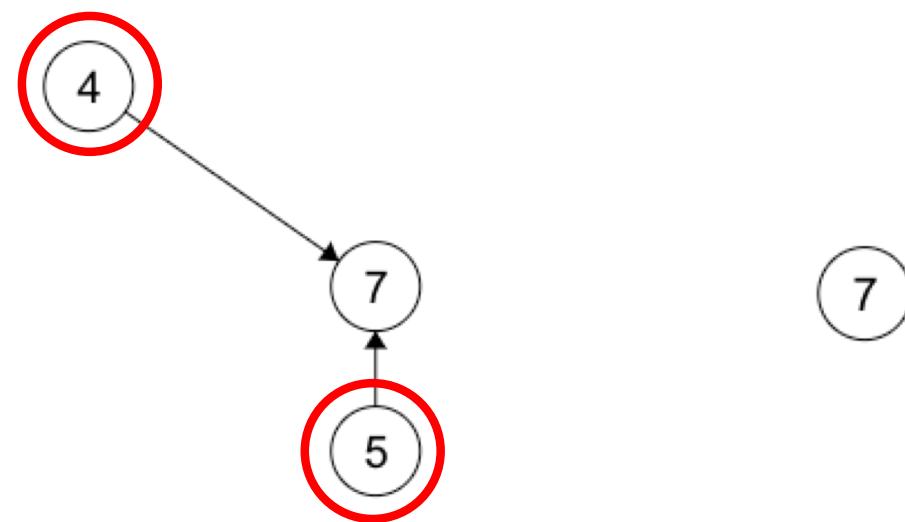
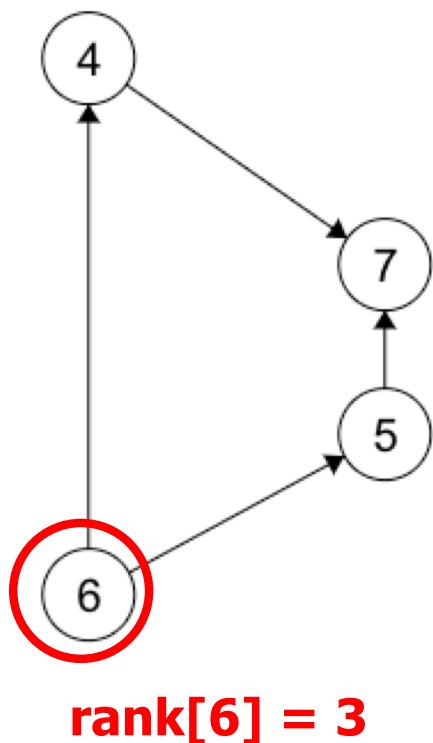
Xếp hạng các đỉnh của đồ thị



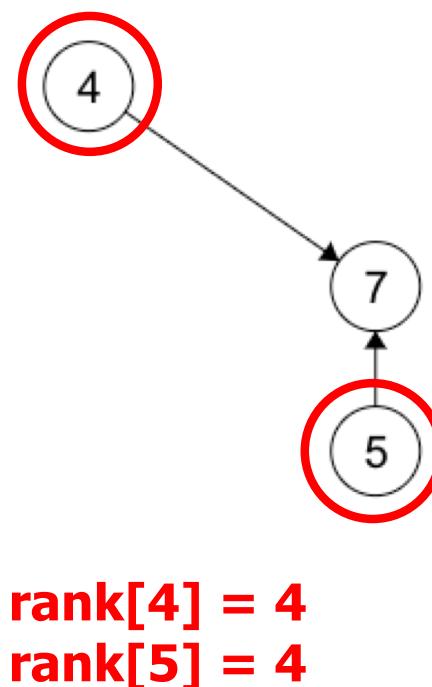
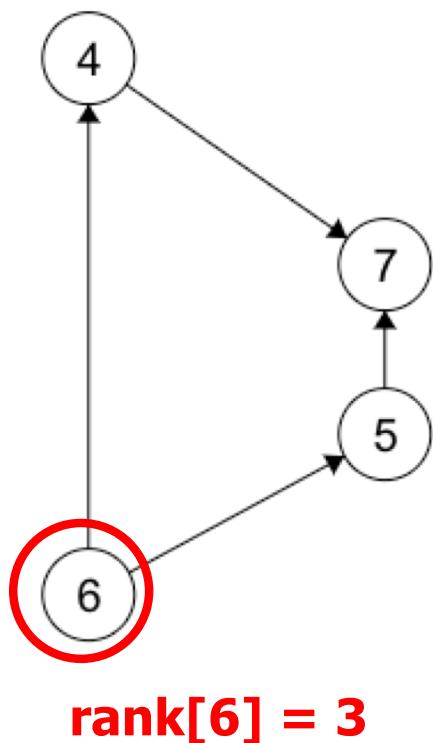
rank[6] = 3



Xếp hạng các đỉnh của đồ thị



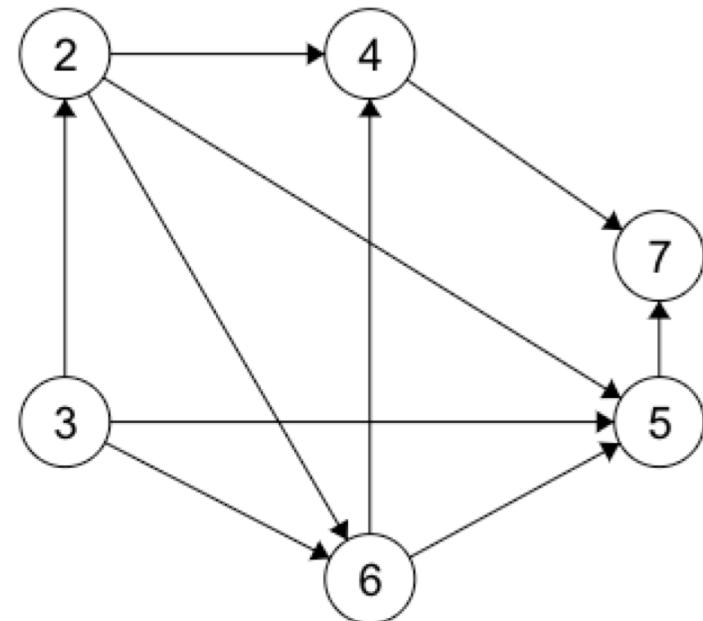
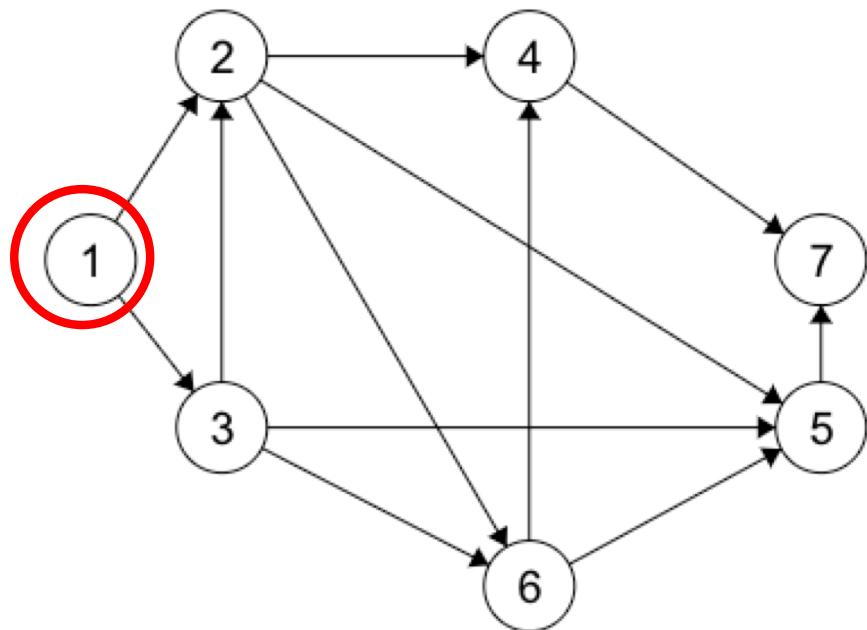
Xếp hạng các đỉnh của đồ thị



rank[7] = 5

Xếp hạng các đỉnh của đồ thị

- Nhận xét:
 - Khi xoá 1 đỉnh gốc,
 - có gì thay đổi đối với đồ thị?
 - đỉnh nào sẽ trở thành gốc mới?

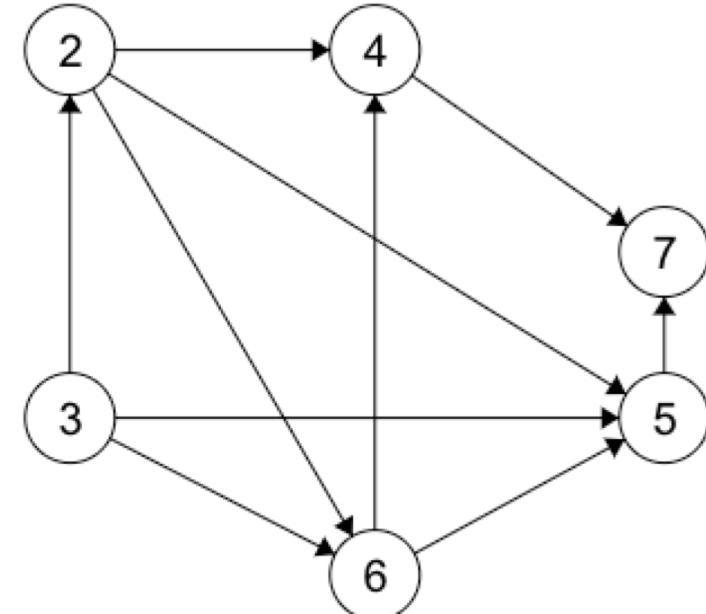
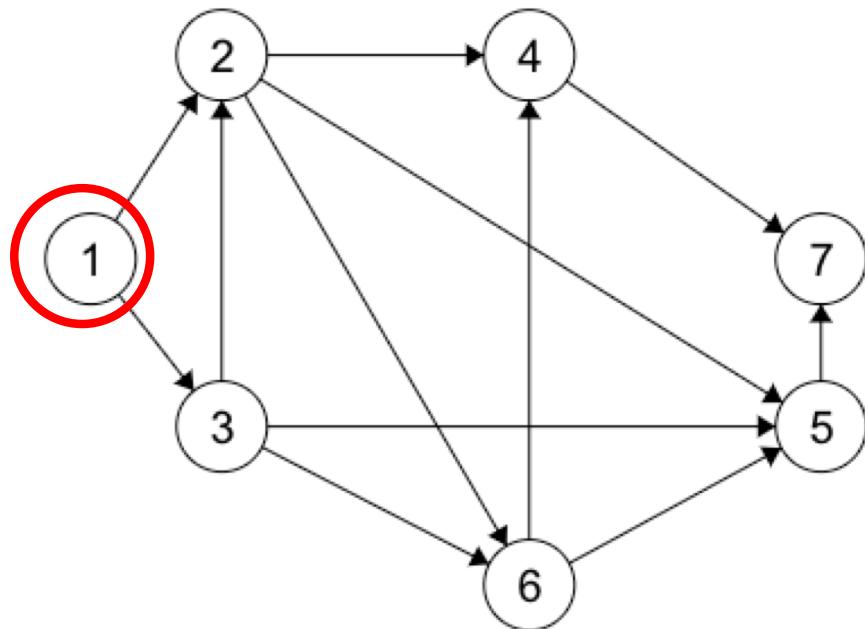


Xếp hạng các đỉnh của đồ thị

- Nhận xét:

- Khi xoá 1 đỉnh gốc,
 - có gì thay đổi đối với đồ thị?
 - đỉnh nào sẽ trở thành gốc mới?

- Đỉnh gốc mất đi, các cung liên thuộc với nó cũng mất đi.
- Những đỉnh nào sẽ ảnh hưởng?

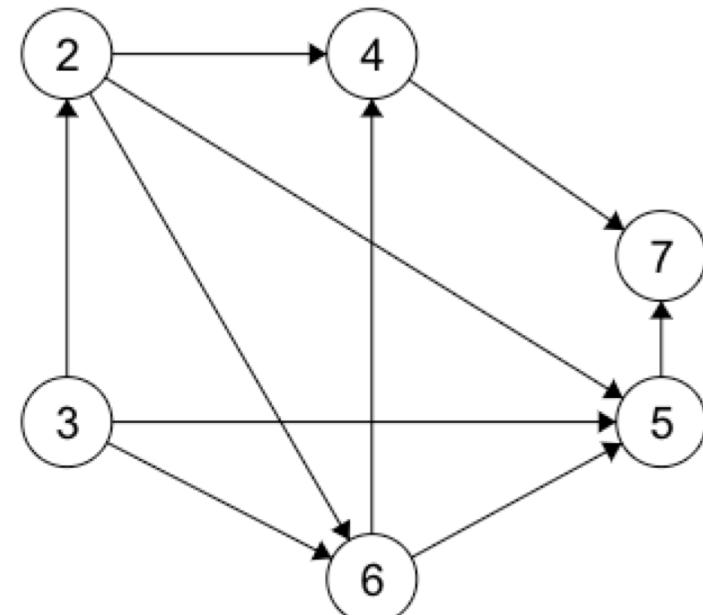
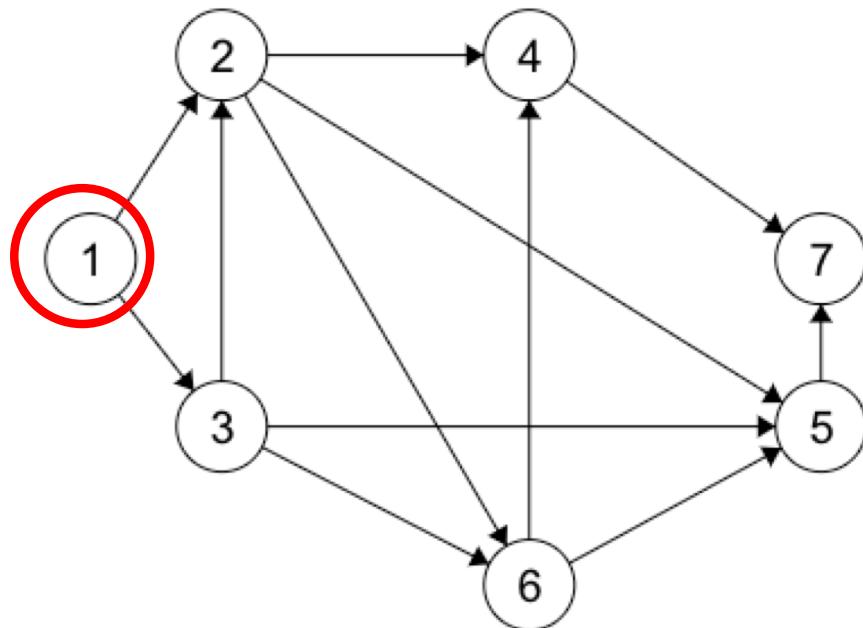


Xếp hạng các đỉnh của đồ thị

- Nhận xét:

- Khi xoá 1 đỉnh gốc,
 - có gì thay đổi đối với đồ thị?
 - đỉnh nào sẽ trở thành gốc mới?

- Đỉnh gốc mất đi, các cung liên thuộc với nó cũng mất đi.
- Những đỉnh nào sẽ ảnh hưởng?
 - Các đỉnh kề của gốc
 - Thay đổi gì?

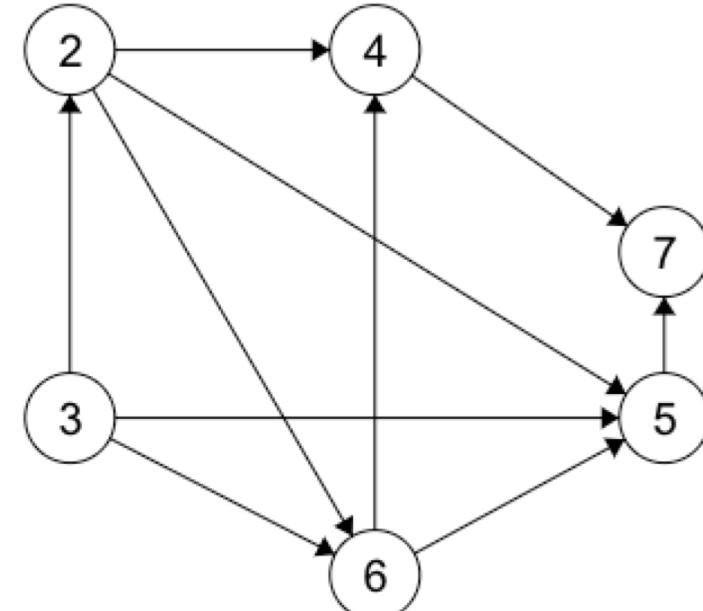
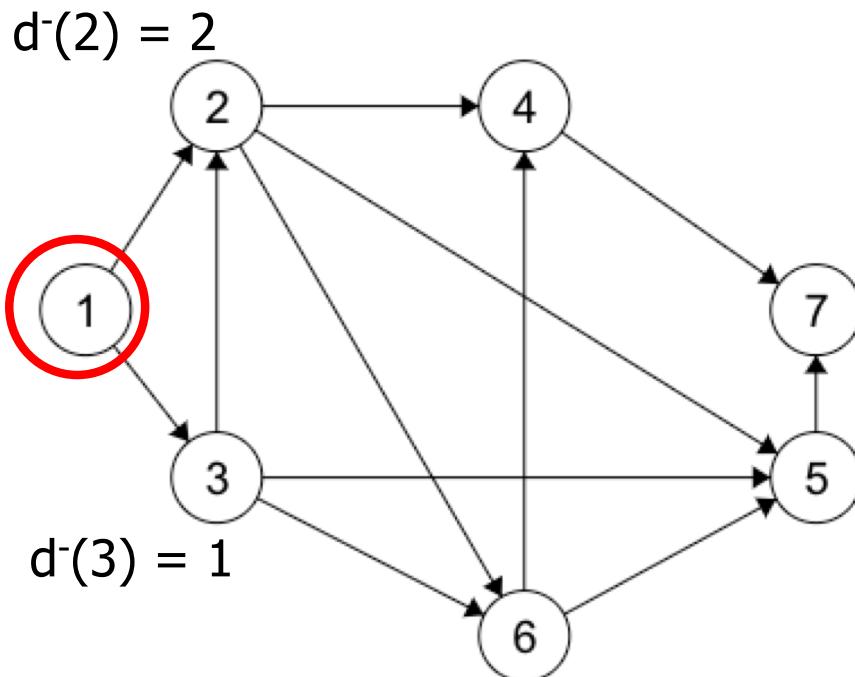


Xếp hạng các đỉnh của đồ thị

- Nhận xét:

- Khi xoá 1 đỉnh gốc,
 - có gì thay đổi đối với đồ thị?
 - đỉnh nào sẽ trở thành gốc mới?

- Đỉnh gốc mất đi, các cung liên thuộc với nó cũng mất đi.
- Những đỉnh nào sẽ ảnh hưởng?
 - Các đỉnh kề của gốc
 - Thay đổi gì?

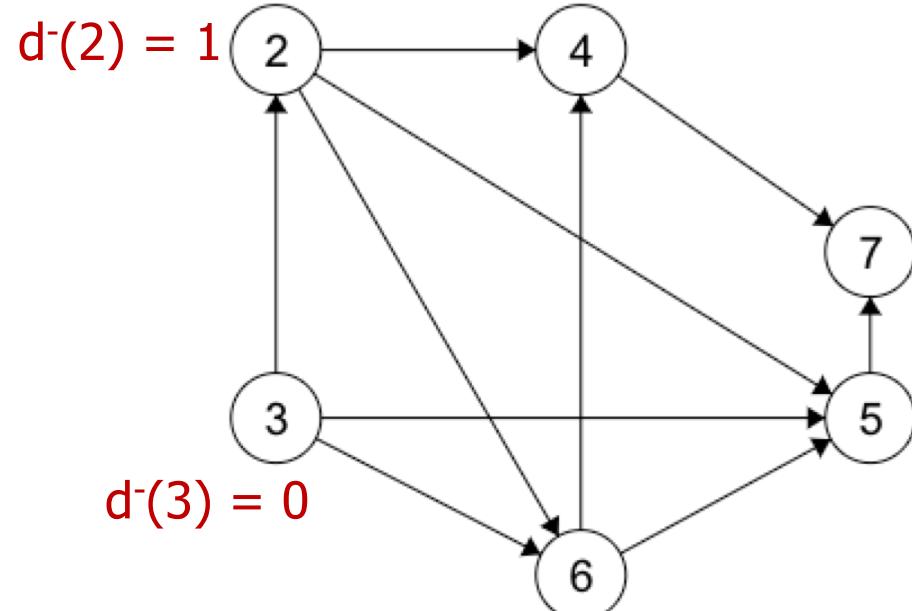
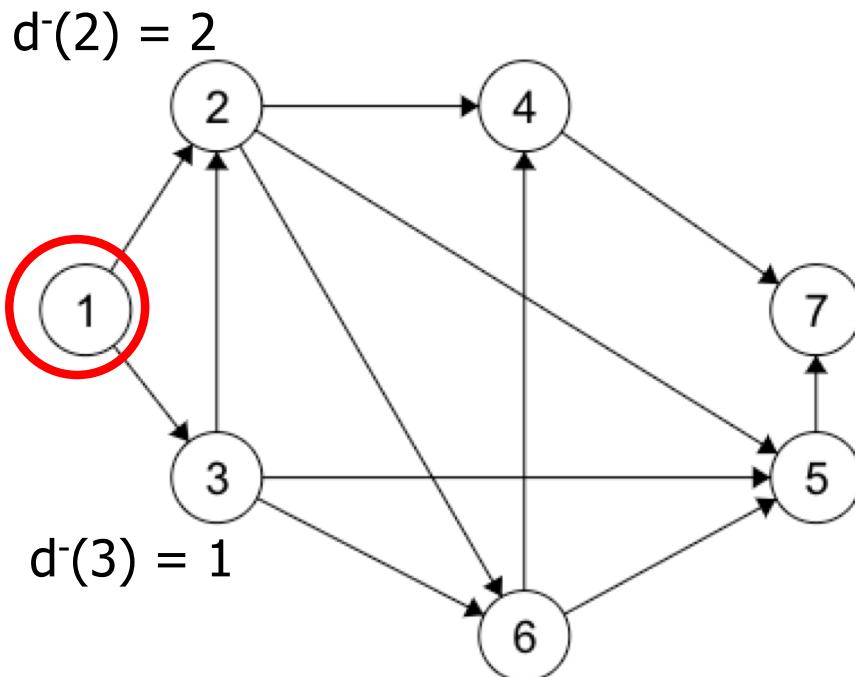


Xếp hạng các đỉnh của đồ thị

- Nhận xét:

- Khi xoá 1 đỉnh gốc,
 - có gì thay đổi đối với đồ thị?
 - đỉnh nào sẽ trở thành gốc mới?

- Đỉnh gốc mất đi, các cung liên thuộc với nó cũng mất đi.
- Những đỉnh nào sẽ ảnh hưởng?
 - Các đỉnh kề của gốc
 - Thay đổi gì?



Xếp hạng các đỉnh của đồ thị

- Nhận xét:
 - Xoá đỉnh u: **giảm bậc vào** của các đỉnh kề của u
 - Đỉnh kề nào có bậc vào giảm về 0 => gốc mới

Xếp hạng các đỉnh của đồ thị

- Nhận xét:
 - Xoá đỉnh u: **giảm bậc vào** của các đỉnh kề của u
 - Đỉnh kề nào có bậc vào giảm về 0 => gốc mới
- Cài đặt
 - Các biến hỗ trợ
 - **d[u]**: bậc vào của đỉnh u
 - **S[k]**: danh sách các gốc ở bước k
 - **rank[u]**: hạng của đỉnh u
 - **k**: bước lặp

Xếp hạng các đỉnh của đồ thị

Khởi tạo

Tính bậc vào của các đỉnh: $d[u]$

Đưa các gốc vào $S[0]$

$k = 0$

while ($S[k] \neq$ rỗng)

//1. Làm rỗng danh sách $S[k+1]$

//2. Xếp hạng và giảm bậc của đỉnh kề

for (các đỉnh u trong danh sách $S[k]$)

rank[u] = k

for (các đỉnh kề v của u)

$d[v]$ --

if ($d[v] == 0$)

đưa v vào $S[k+1]$

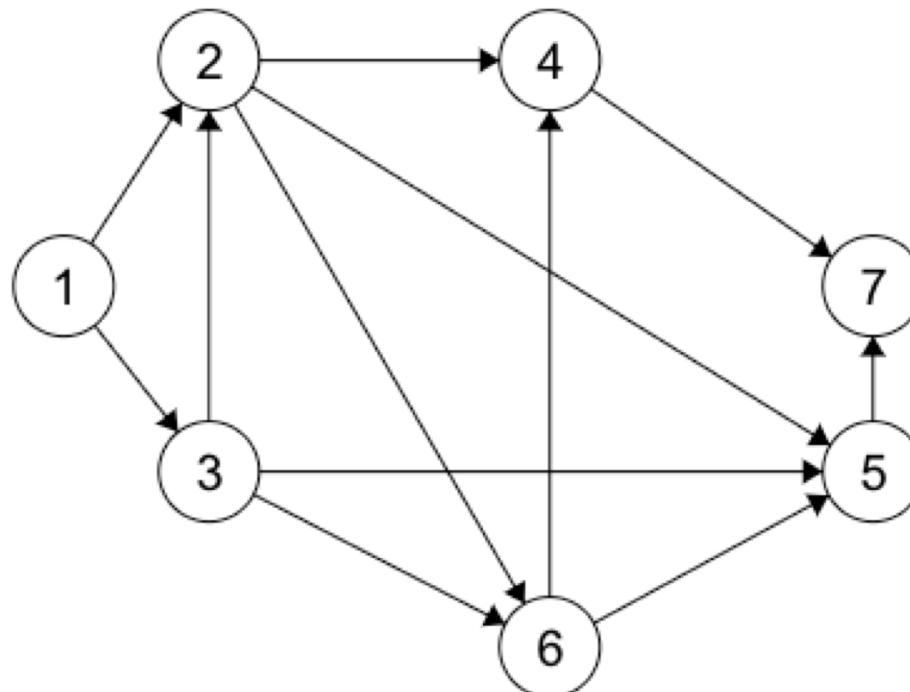
$k++$

xếp hạng cho u

Xoá u (giảm bậc
vào của các đỉnh
kề của u)

Bài tập

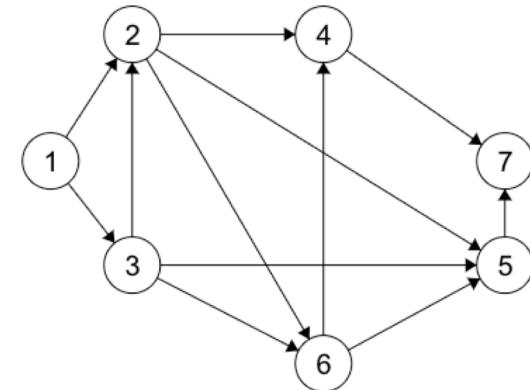
- Áp dụng thuật toán xếp hạng, hãy xếp hạng đồ thị DAG sau:



1 2
1 3
2 4
2 5
2 6
3 2
3 5
3 6
4 7
5 7
6 4
6 5

Bài tập

- Khởi tạo



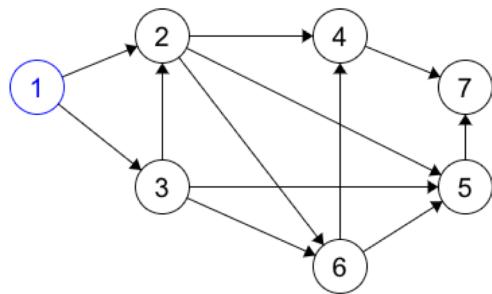
Bước (k)	1	2	3	4	5	6	7	S[k+1]	Công việc
KT (-1)	$d[1] = 0$	$d[2] = 2$	$d[3] = 1$	$d[4] = 2$	$d[5] = 3$	$d[6] = 2$	$d[7] = 2$	{1}	Tính $d[u]$ và $S[0]$
0									
1									
2									

	Help	Clear	shift	Delete	Edit	Undo										
	1	2	3	4	5	6	7	S[k+1]	Công việc							
Khởi tạo																
0																
1																
2																
3																
4																
5																
Kết quả:																

	Help	Clear	shift	Delete	Edit	Undo								
	Áp dụng thuật toán xếp hạng và ghi kết quả vào bảng:													
	1	2	3	4	5	6	7	S[k+1]	Công việc					
Khởi tạo	0	2	1	2	3	2	2	1	Khởi tạo					
0														
1														
2														
3														
4														
5														
Kết quả:														

	Help	Clear	shift	Delete	Edit	Undo								
	1	2	3	4	5	6	7	S[k+1]		Công việc				
Khởi tạo	0		2	1	2	3	2			1			Khởi tạo	
0	*		1	0						3		Xếp hạng 1 C/N: 2, 3		
1														
2														
3														
4														
5														
Kết quả:														

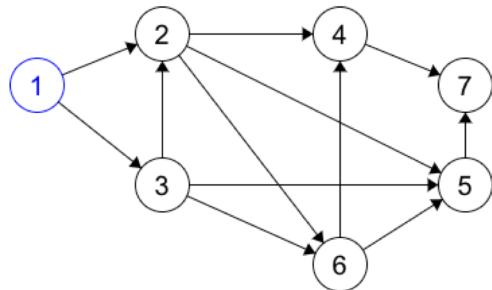
Help Clear shift Delete Edit Undo



Áp dụng thuật toán xếp hạng và ghi kết quả vào bảng:

	1	2	3	4	5	6	7	$S[k+1]$	Công việc
Khởi tạo	0	2	1	2	3	2	2	1	Khởi tạo
0	*	1	0					3	Xếp hạng 1 C/N: 2, 3
1		0	*		2	1		2	Xếp hạng 3 C/N: 2, 5, 6
2									
3									
4									
5									
Kết quả:									

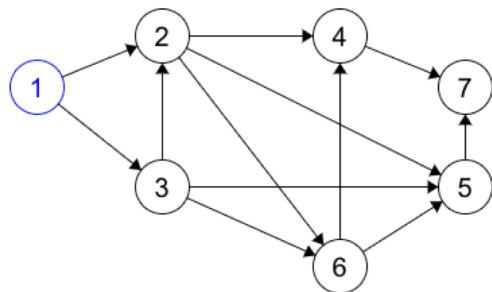
Help Clear shift Delete Edit Undo



Áp dụng thuật toán xếp hạng và ghi kết quả vào bảng:

	1	2	3	4	5	6	7	S[k+1]	Công việc
Khởi tạo	0	2	1	2	3	2	2	1	Khởi tạo
0	*	1	0					3	Xếp hạng 1 C/N: 2, 3
1		0	*		2	1		2	Xếp hạng 3 C/N: 2, 5, 6
2		*		1	1	0		6	Xếp hạng 2 C/N: 4, 5, 6
3									
4									
5									
Kết quả:									

Help Clear shift Delete Edit Undo



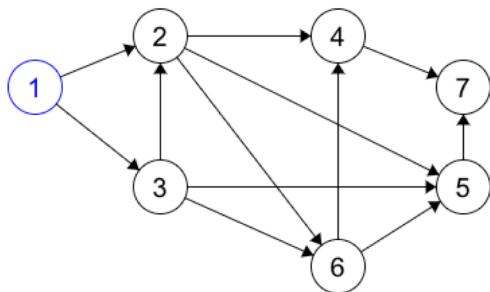
Áp dụng thuật toán xếp hạng và ghi kết quả vào bảng:

	1	2	3	4	5	6	7	$S[k+1]$	Công việc
Khởi tạo	0	2	1	2	3	2	2	1	Khởi tạo
0	*	1	0					3	Xếp hạng 1 C/N: 2, 3
1		0	*		2	1		2	Xếp hạng 3 C/N: 2, 5, 6
2		*		1	1	0		6	Xếp hạng 2 C/N: 4, 5, 6
3				0	0	*		5,4	Xếp hạng 6 C/N: 4, 5
4									
5									
Kết quả:									

	Help	Clear	shift	Delete	Edit	Undo							
	1	2	3	4	5	6	7	S[k+1]	Công việc				
Khởi tạo	0		2	1	2	3	2	1					Khởi tạo
0	*		1	0				3					Xếp hạng 1 C/N: 2, 3
1		0	*		2	1					2		Xếp hạng 3 C/N: 2, 5, 6
2		*			1	1	0				6		Xếp hạng 2 C/N: 4, 5, 6
3				0	0	*				5,4			Xếp hạng 6 C/N: 4, 5
4				*	*			0		7			Xếp hạng 4, 5 C/N: 7
5													
Kết quả:													

	Help	Clear	shift	Delete	Edit	Undo							
	1	2	3	4	5	6	7	S[k+1]	Công việc				
Khởi tạo	0		2	1	2	3	2	1		Khởi tạo			
0	*		1	0				3		Xếp hạng 1 C/N: 2, 3			
1		0	*		2	1		2		Xếp hạng 3 C/N: 2, 5, 6			
2		*			1	1	0		6	Xếp hạng 2 C/N: 4, 5, 6			
3				0	0	*		5,4		Xếp hạng 6 C/N: 4, 5			
4				*	*		0	7		Xếp hạng 4, 5 C/N: 7			
5							*			Xếp hạng 7 C/N:			
Kết quả:													

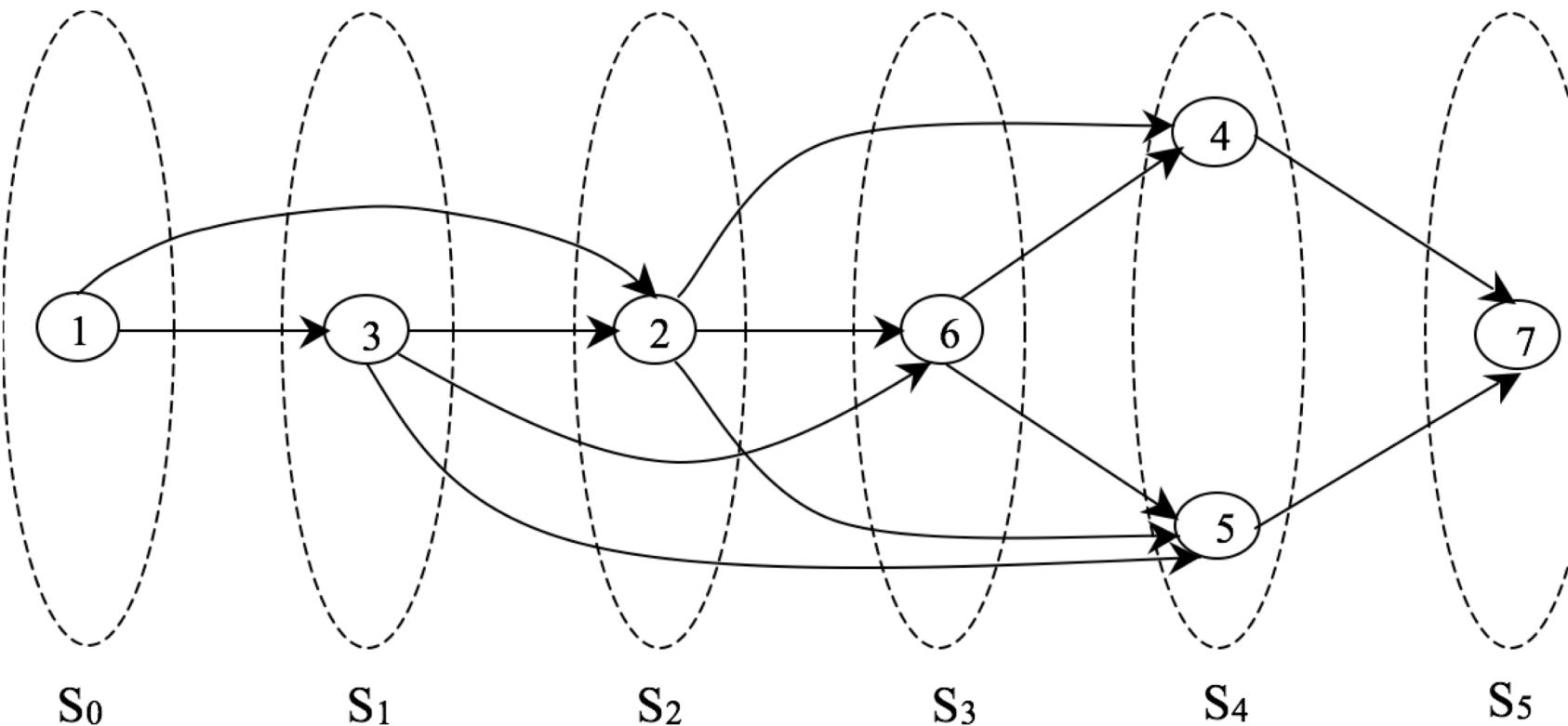
Help Clear shift Delete Edit Undo



Áp dụng thuật toán xếp hạng và ghi kết quả vào bảng:

	1	2	3	4	5	6	7	S[k+1]	Công việc
Khởi tạo	0	2	1	2	3	2	2	1	Khởi tạo
0	*	1	0					3	Xếp hạng 1 C/N: 2, 3
1		0	*		2	1		2	Xếp hạng 3 C/N: 2, 5, 6
2		*		1	1	0		6	Xếp hạng 2 C/N: 4, 5, 6
3				0	0	*		5,4	Xếp hạng 6 C/N: 4, 5
4				*	*		0	7	Xếp hạng 4, 5 C/N: 7
5							*		Xếp hạng 7 C/N:
Kết quả:	0	2	1	4	4	3	5		Kết quả

Bài tập



Thứ tự topo: 1, 3, 2, 6, 4, 5, 7
1, 3, 2, 6, 5, 4, 7

Quản lý dự án



Quản lý dự án

- Dự án (project)
 - Danh sách *công việc* (activities/tasks), mỗi công việc có thời gian hoàn thành
 - Có *sự phụ thuộc giữa các công việc* (vd: công việc B chỉ có thể bắt đầu làm khi làm xong công việc A)
- Vấn đề
 - Kế hoạch thực hiện các công việc
 - Ước tính thời gian hoàn thành

Quản lý dự án

- Dự án (project)

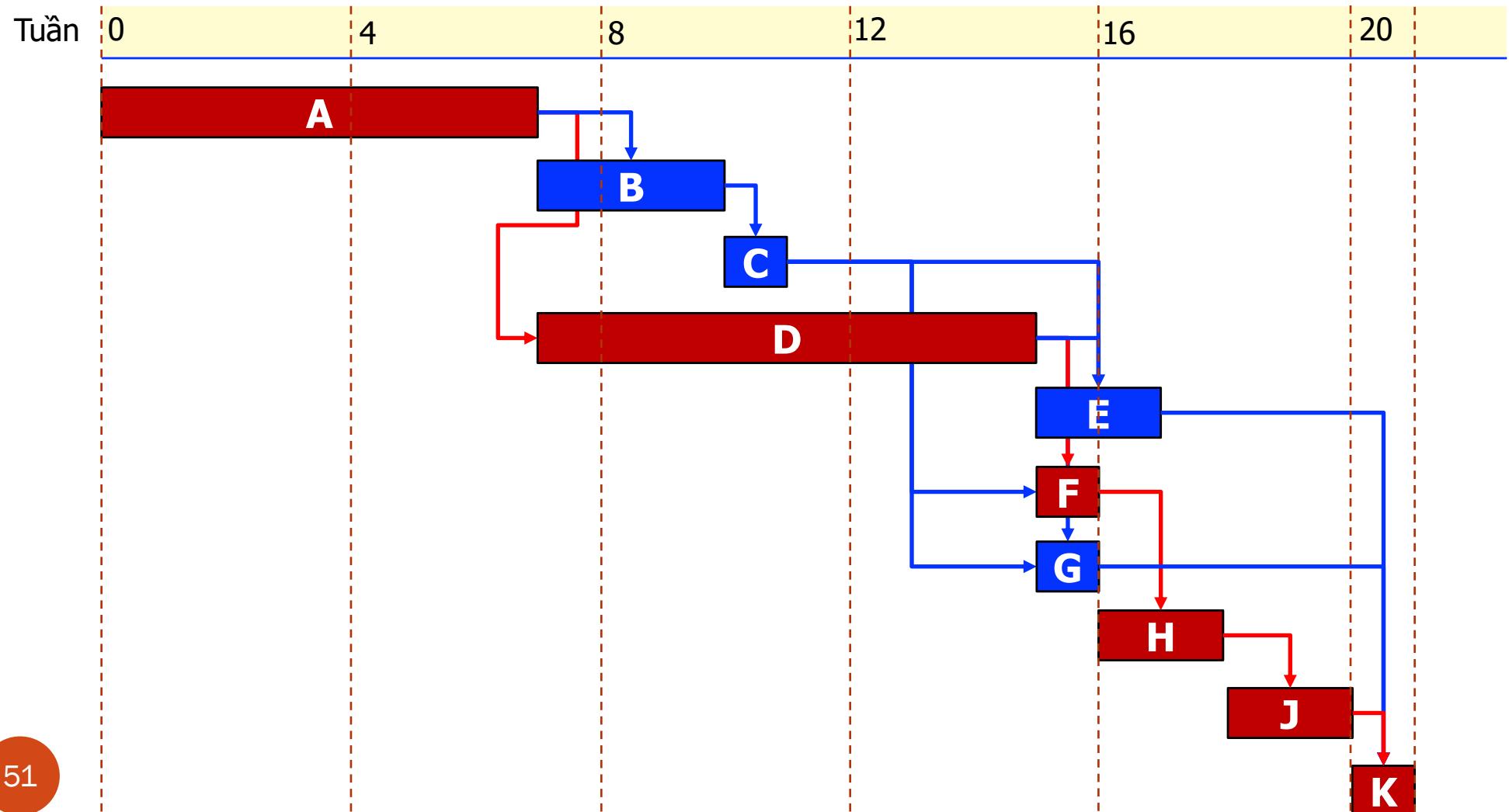
Công việc	Nội dung công việc	Thời gian thực hiện d(i) tính theo tuần	Công việc trước đó
A	Các công việc hồ (nè)	7	Ø
B	Dựng khung cho mái	3	A
C	Lợp mái	1	B
D	Lắp đặt hệ thống vệ sinh, chiếu sáng	8	A
E	Trang trí mặt tiền	2	C, D
F	Ráp cửa sổ	1	C, D
G	Trang hoàng vườn	1	C, D
H	Làm trần	2	F
J	Sơn phết	2	H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



Microsoft Project

Quản lý dự án

- Sơ đồ GANTT



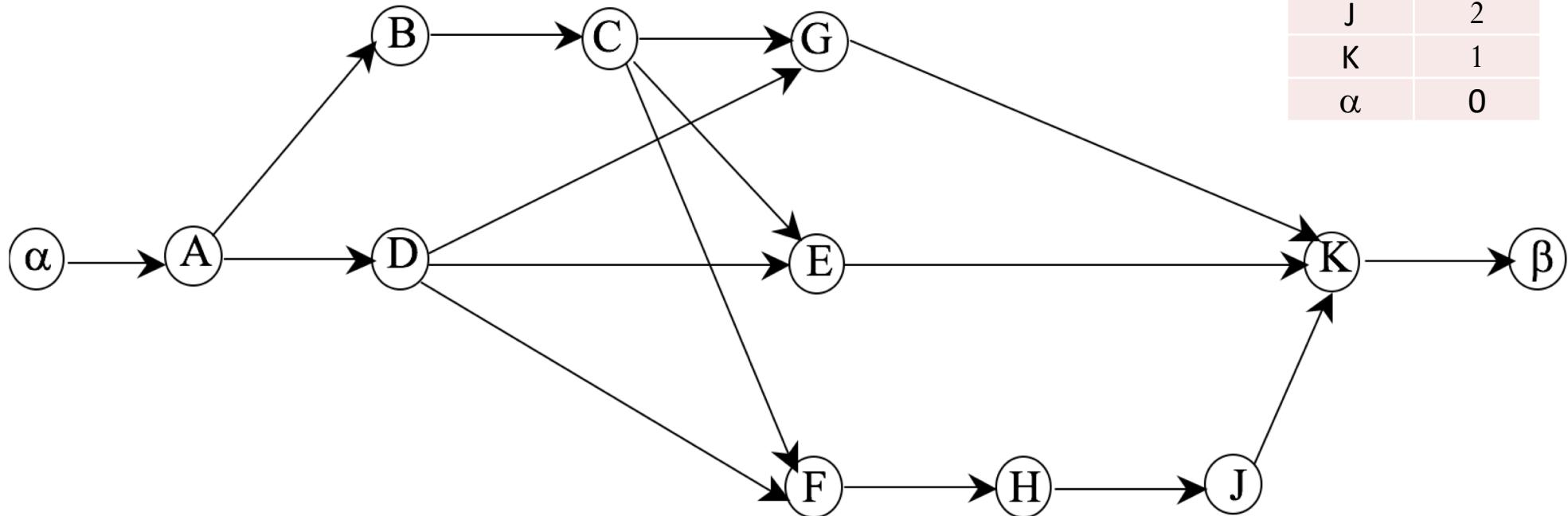
Quản lý dự án

- Mô hình hoá về đồ thị
 - Công việc (CV) => Đỉnh
 - Sư phụ thuộc giữa 2 CV => Cung
 - Thời gian hoàn thành CV u ($d[u]$) => trọng số
của đỉnh u
- Thêm 2 đỉnh: α , β (tương ứng với 2 công việc giả)
- Thêm cung nối α với các đỉnh có bậc vào bằng 0
 - Thời gian hoàn thành α : $d[\alpha] = 0$
- Thêm cung nối các đỉnh có bậc ra bằng 0 với β

Quản lý dự án

- Người quản lý dự án (project manager)
 - Tổ chức thực hiện công việc như thế nào để
 - Thoả mãn sự phụ thuộc của các công việc
 - Thời gian hoàn thành dự án ít nhất

Quản lý dự án



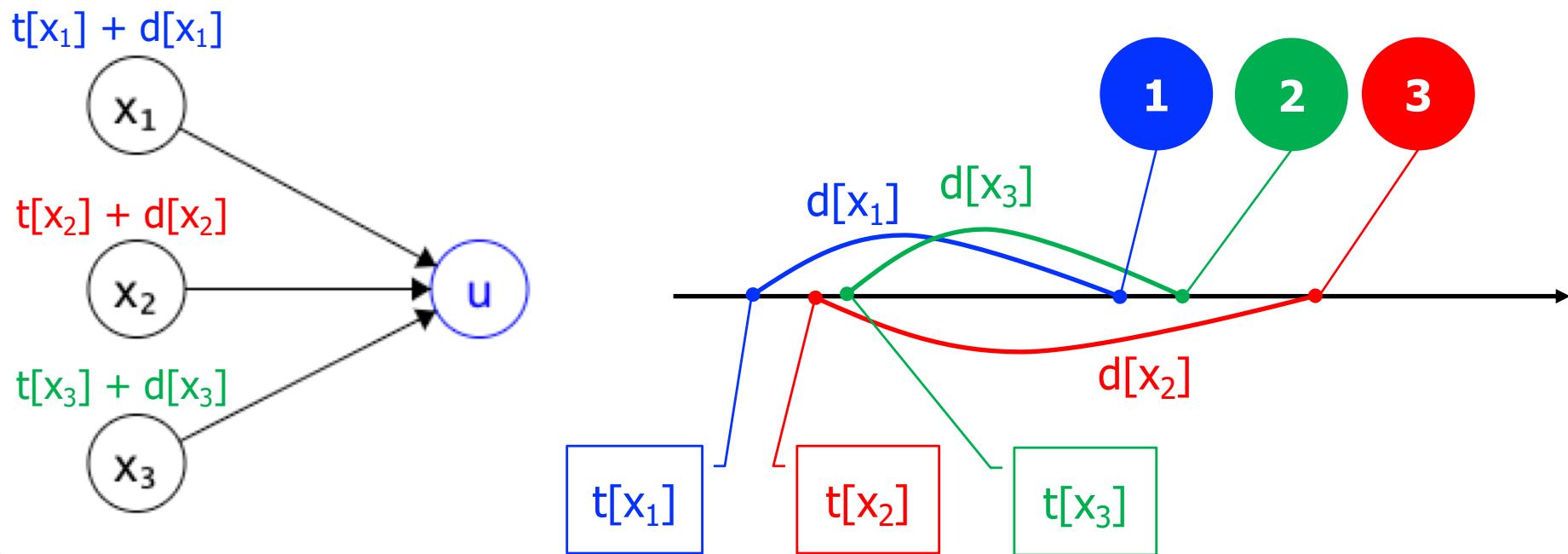
A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

- *Thời điểm bắt đầu sớm nhất* (early start time) của công việc, ký hiệu là $t[u]$
 - Thời điểm sớm nhất có thể bắt đầu công việc u
 - $t[\beta]$: thời điểm sớm nhất hoàn thành dự án
- Tính $t[u]$
 - $t[\alpha] = 0$
 - Xét các đỉnh u theo thứ tự hạng **tăng dần** (theo thứ tự topo)
 - $t[u] = \max \{t[x] + d[x]\}$ với x là đỉnh tương ứng với công việc trước của cung việc u

Quản lý dự án

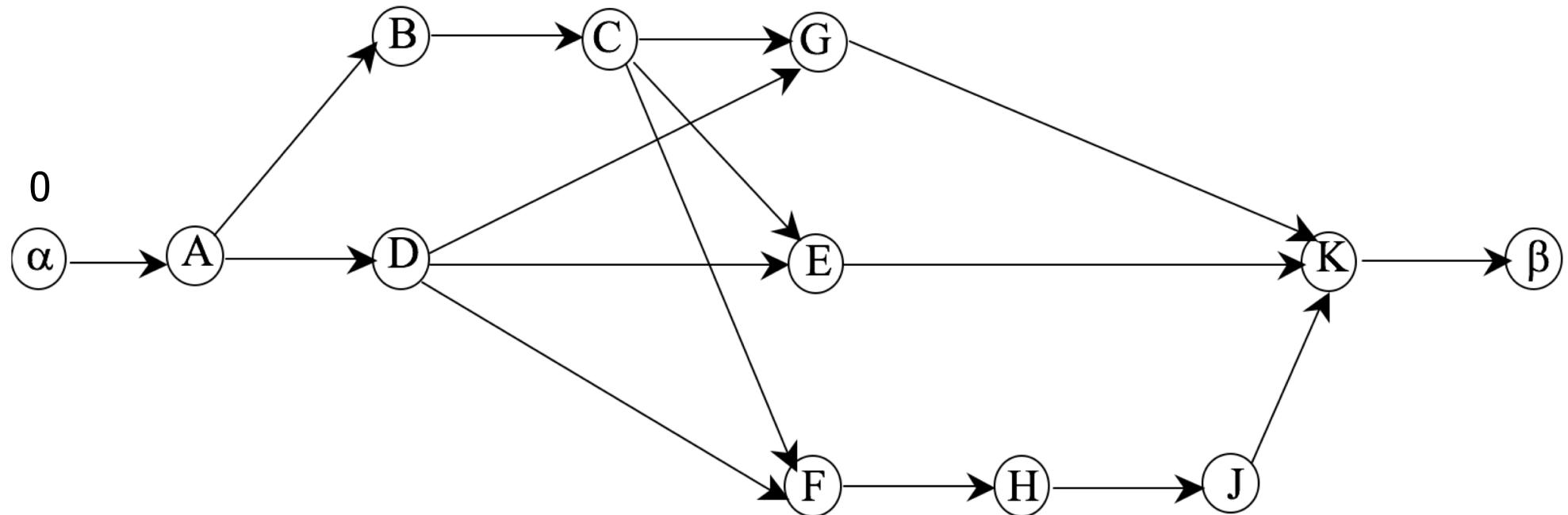
- Tính $t[u]$
 - Xét các đỉnh u theo thứ tự hạng tăng dần
 - $t[u] = \max \{t[x] + d[x]\}$ với x là đỉnh tương ứng với công việc trước của công việc u



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

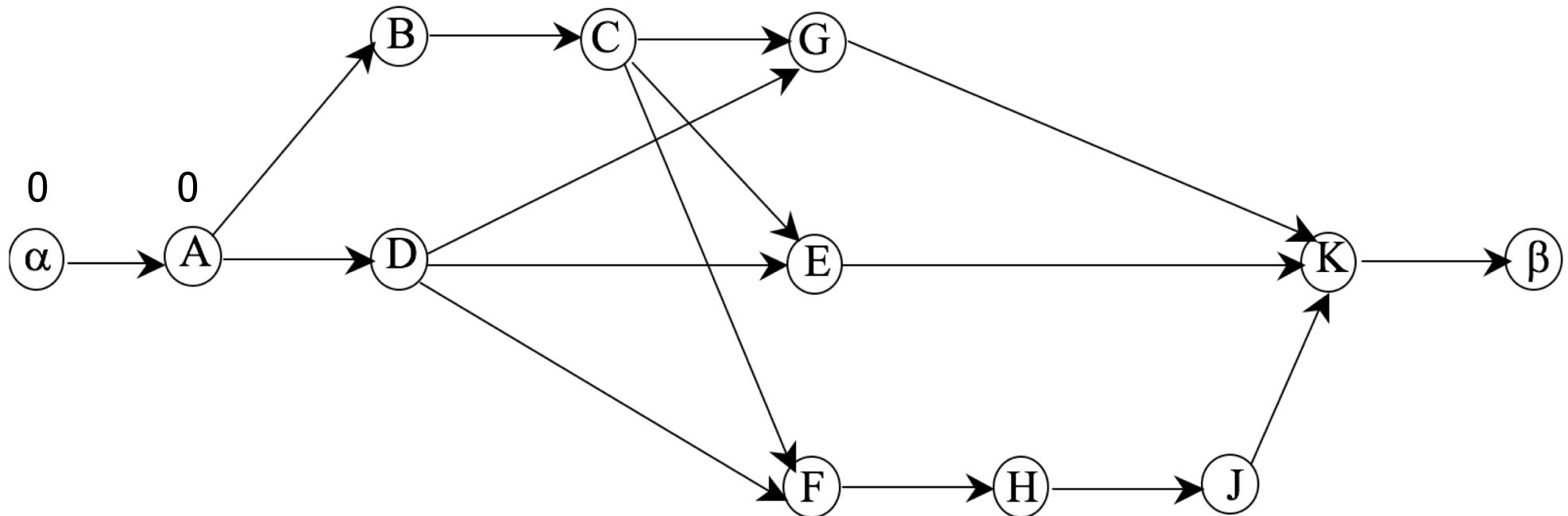
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

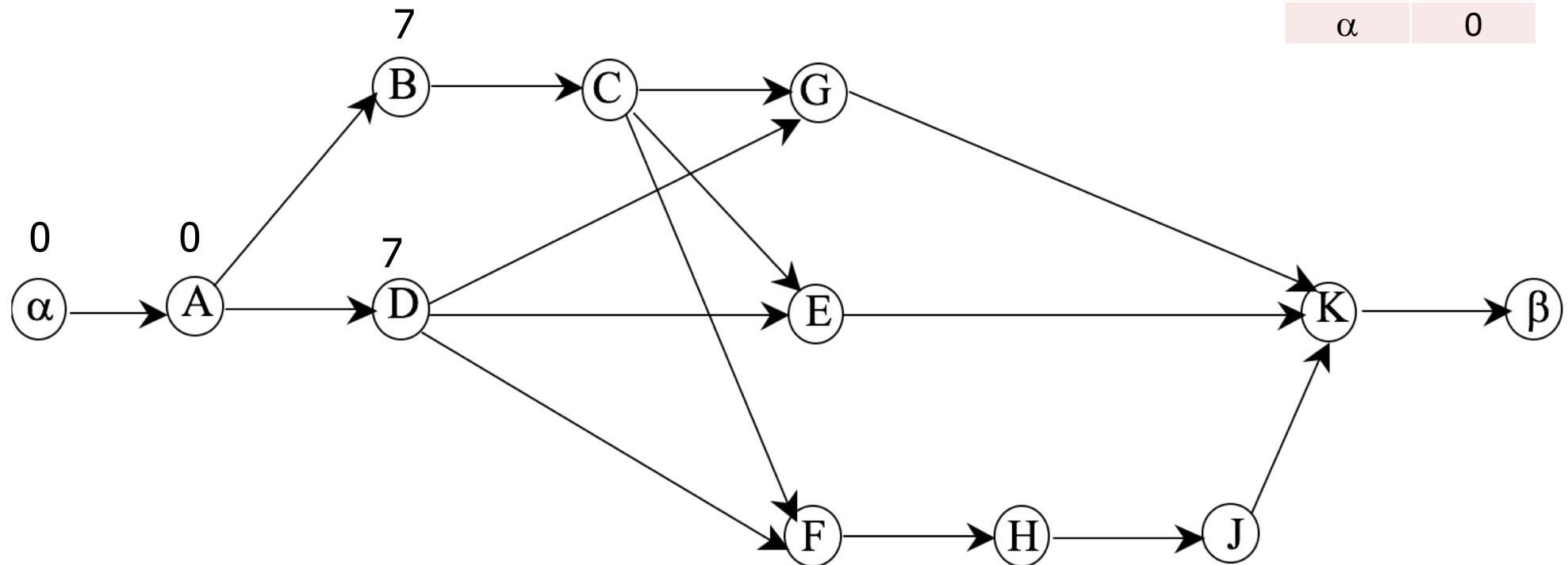
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

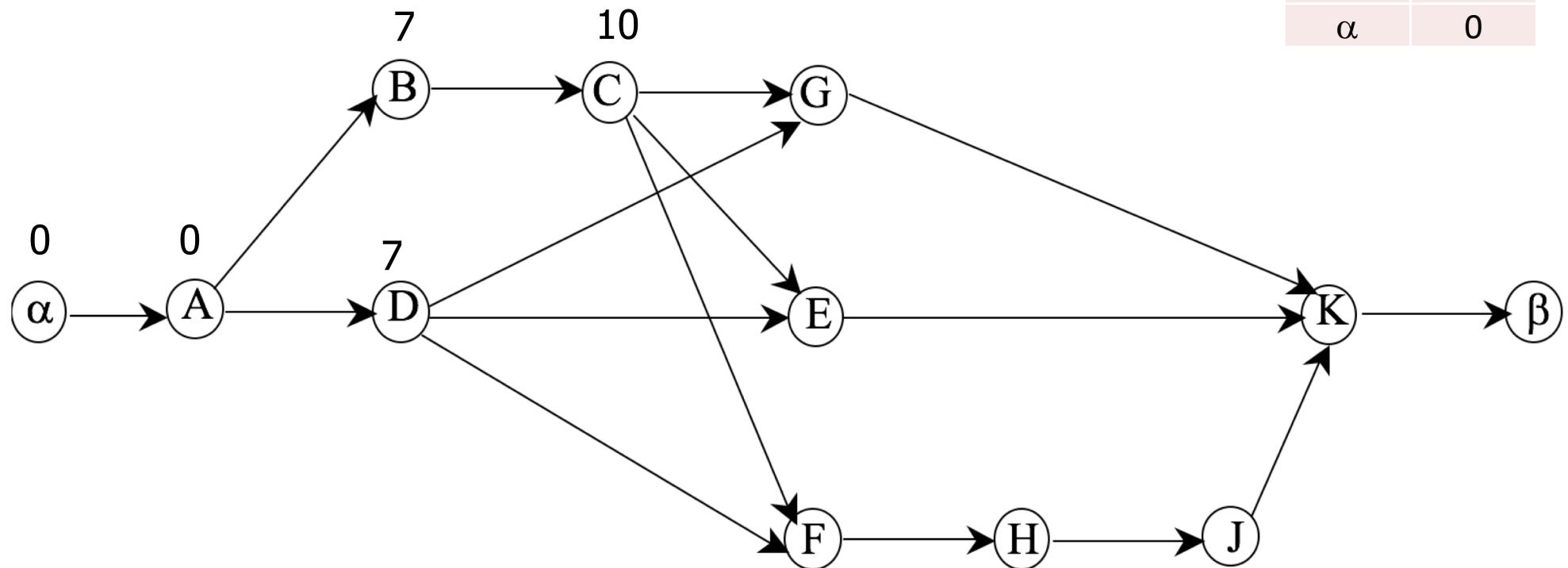
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

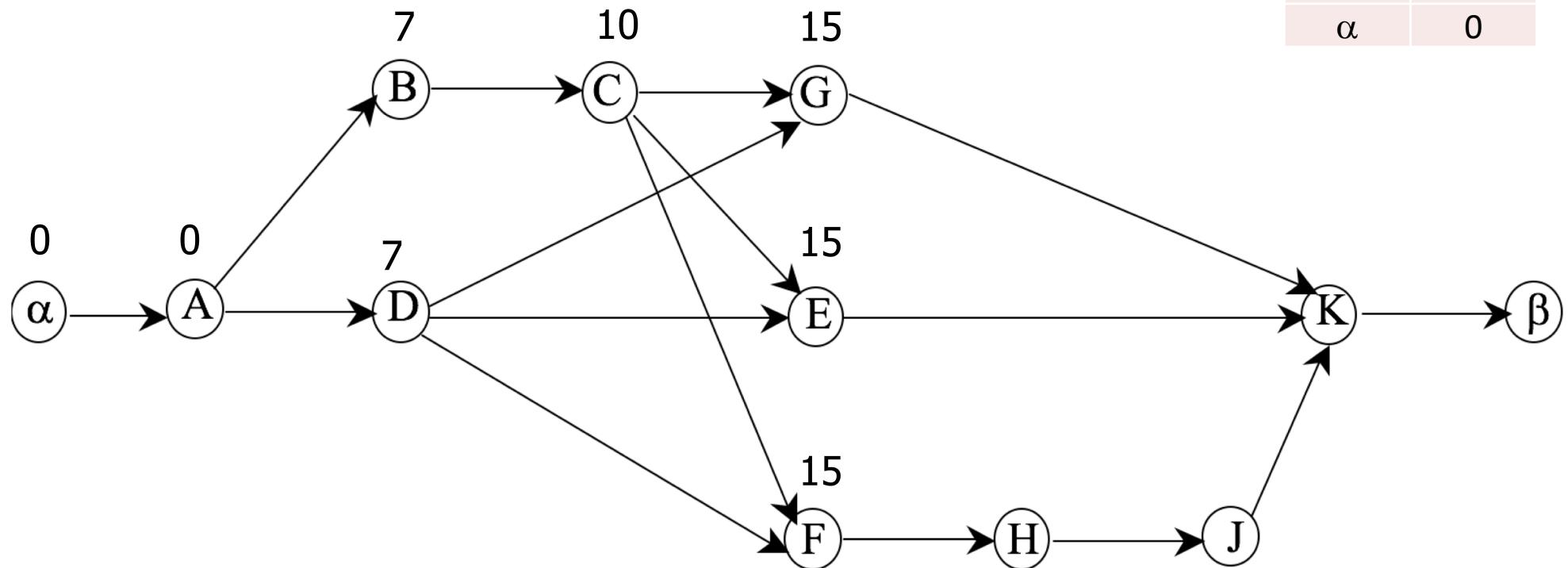
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

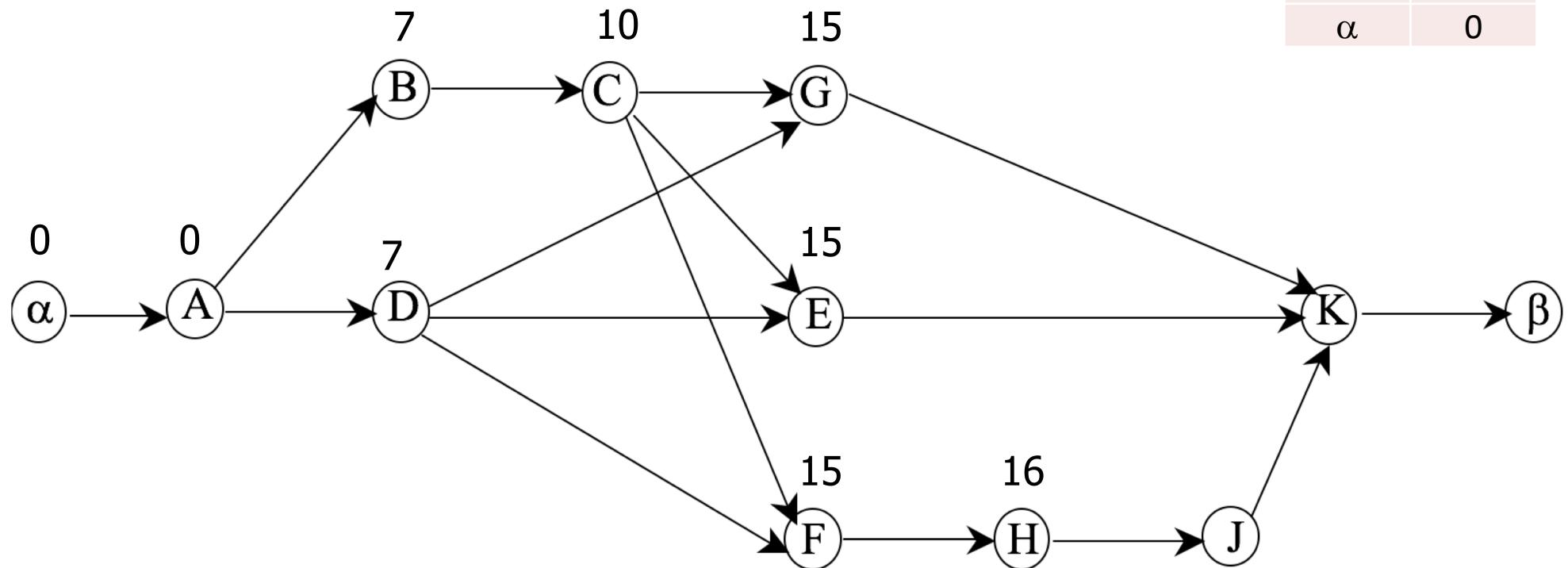
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

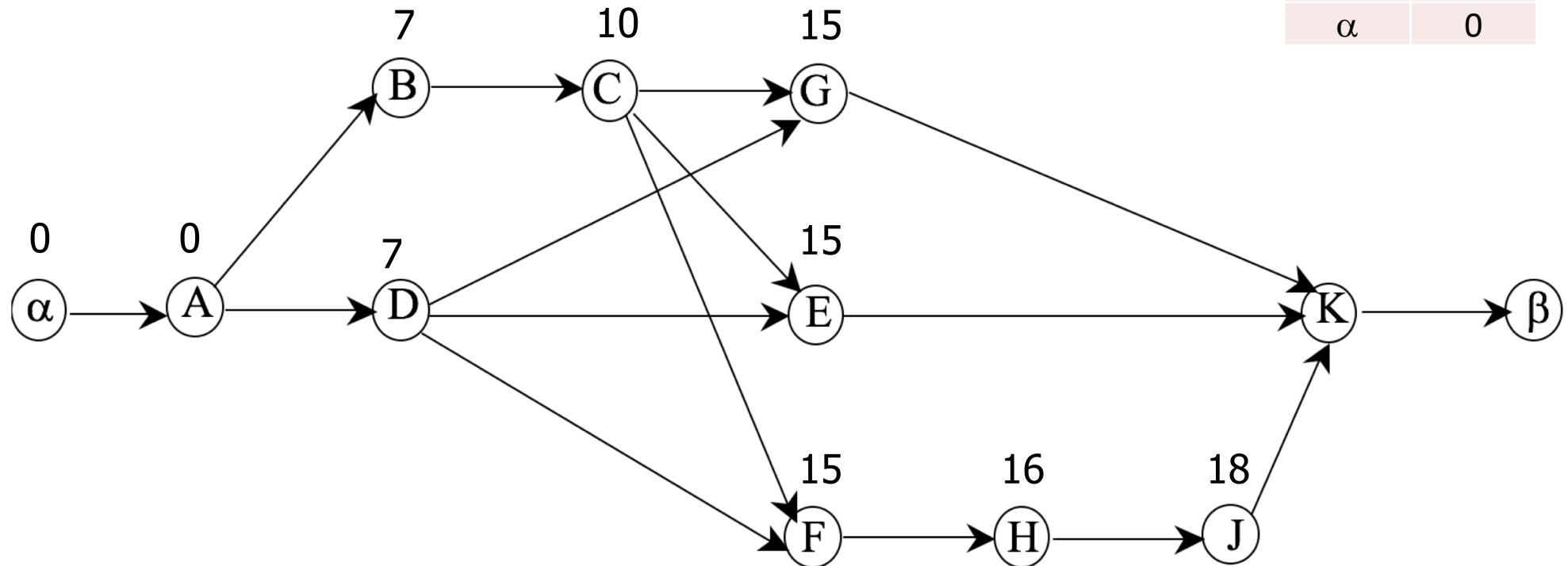
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

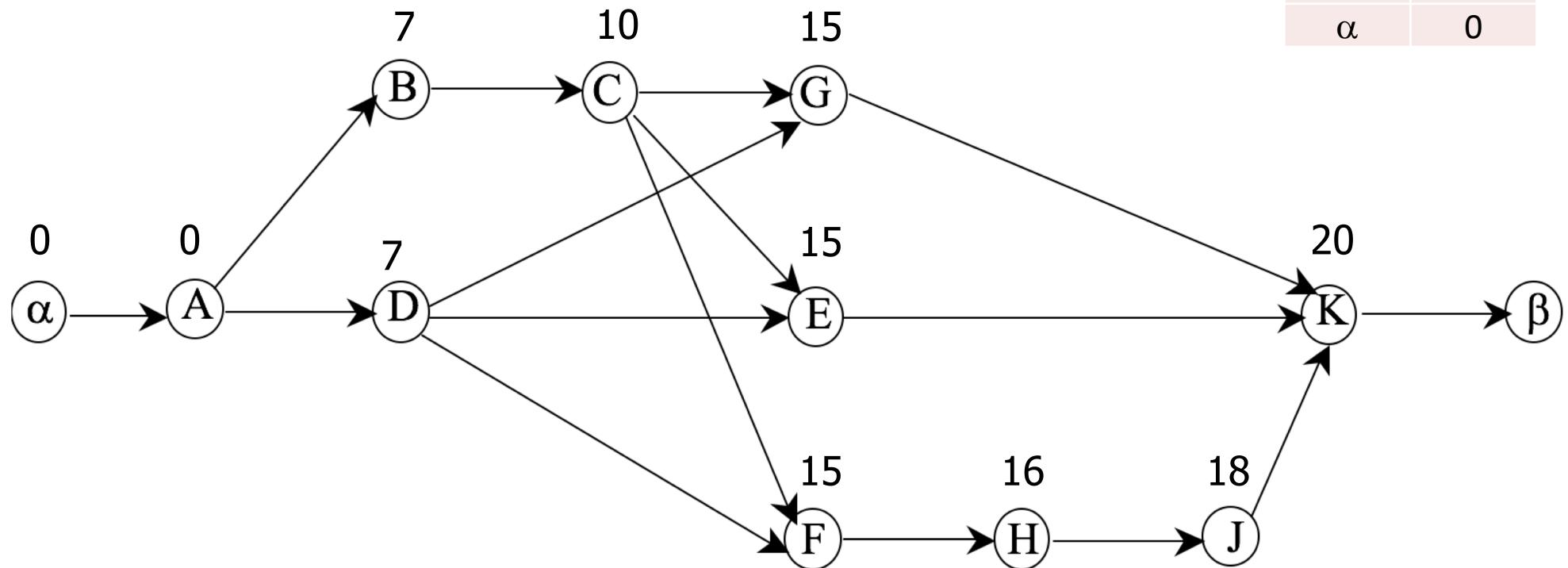
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

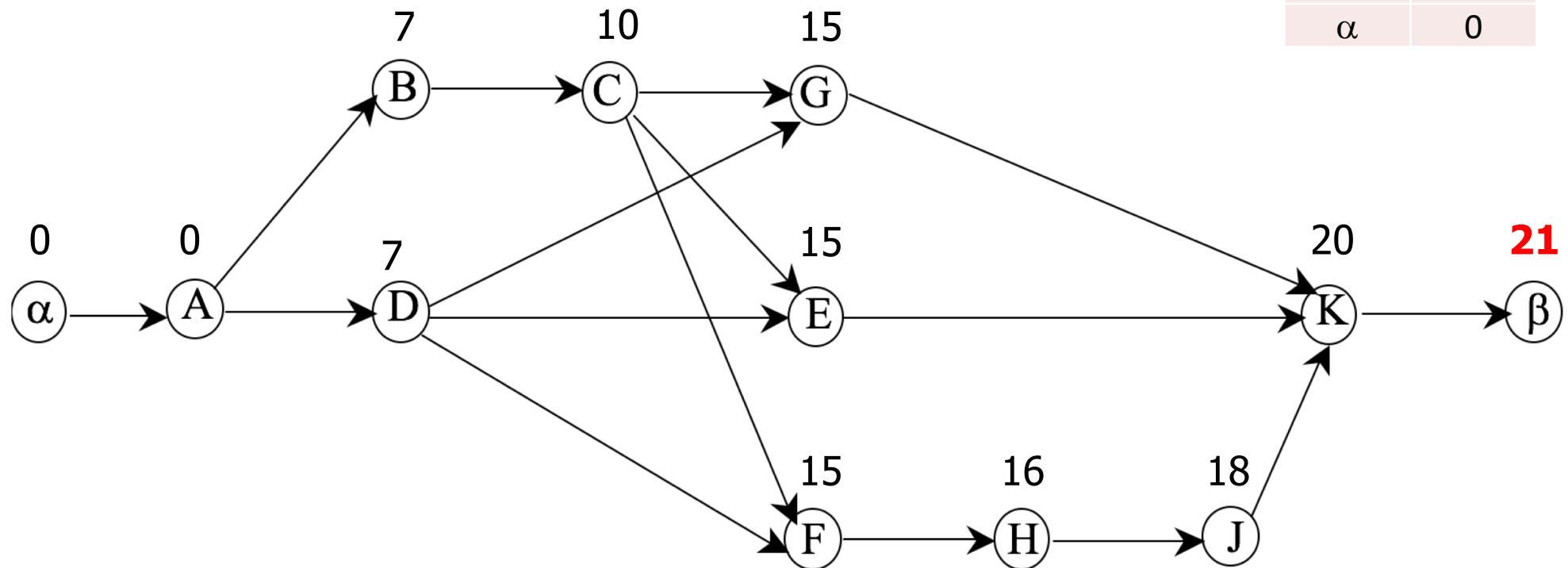
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $t[u]$

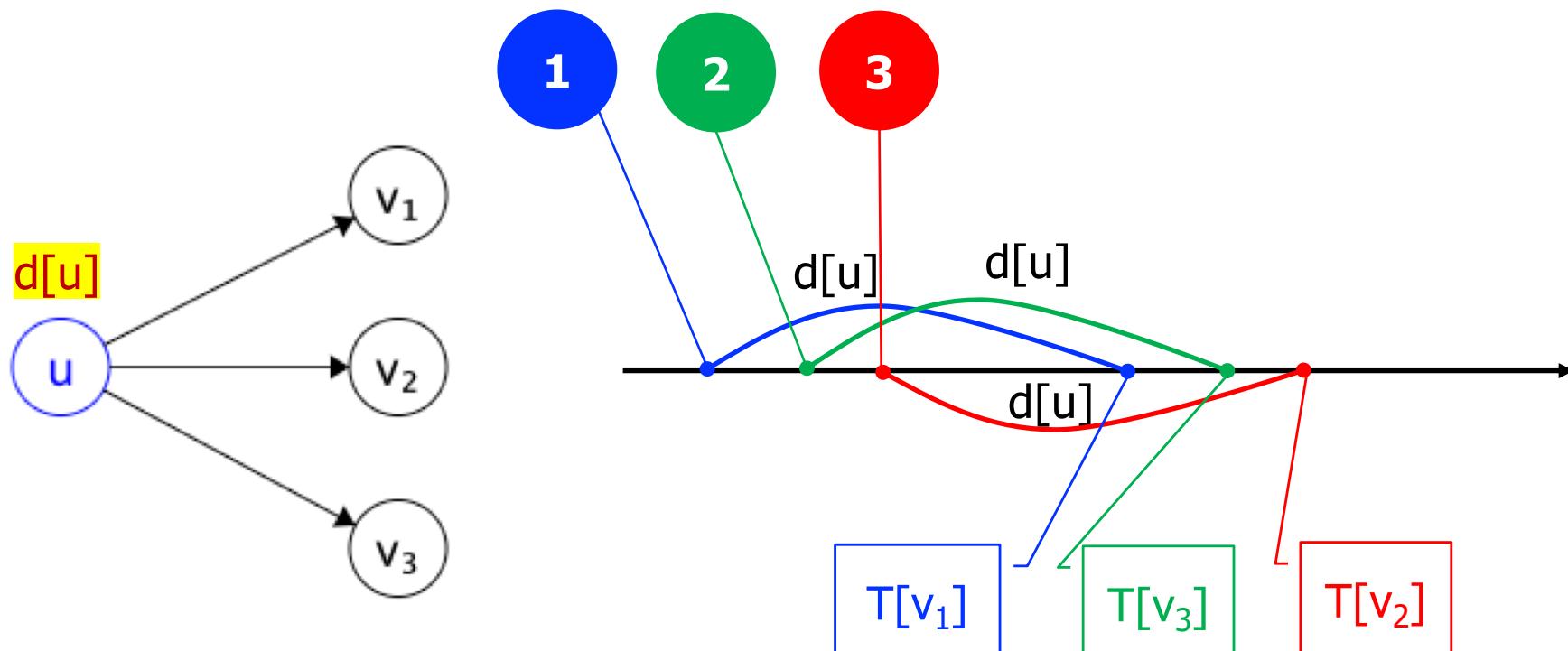


Quản lý dự án

- *Thời điểm bắt đầu trễ nhất* (late start time) của công việc u, ký hiệu $T[u]$
 - Thời điểm trễ nhất để bắt đầu công việc u mà không ảnh hưởng đến tiến độ dự án
- Tính $T[u]$
 - $T[\beta] = t[\beta]$
 - Xét các đỉnh u theo thứ tự hạng **giảm dần**
 - $T[u] = \min \{T[v]\} - d[u]$ với v là đỉnh kề của u

Quản lý dự án

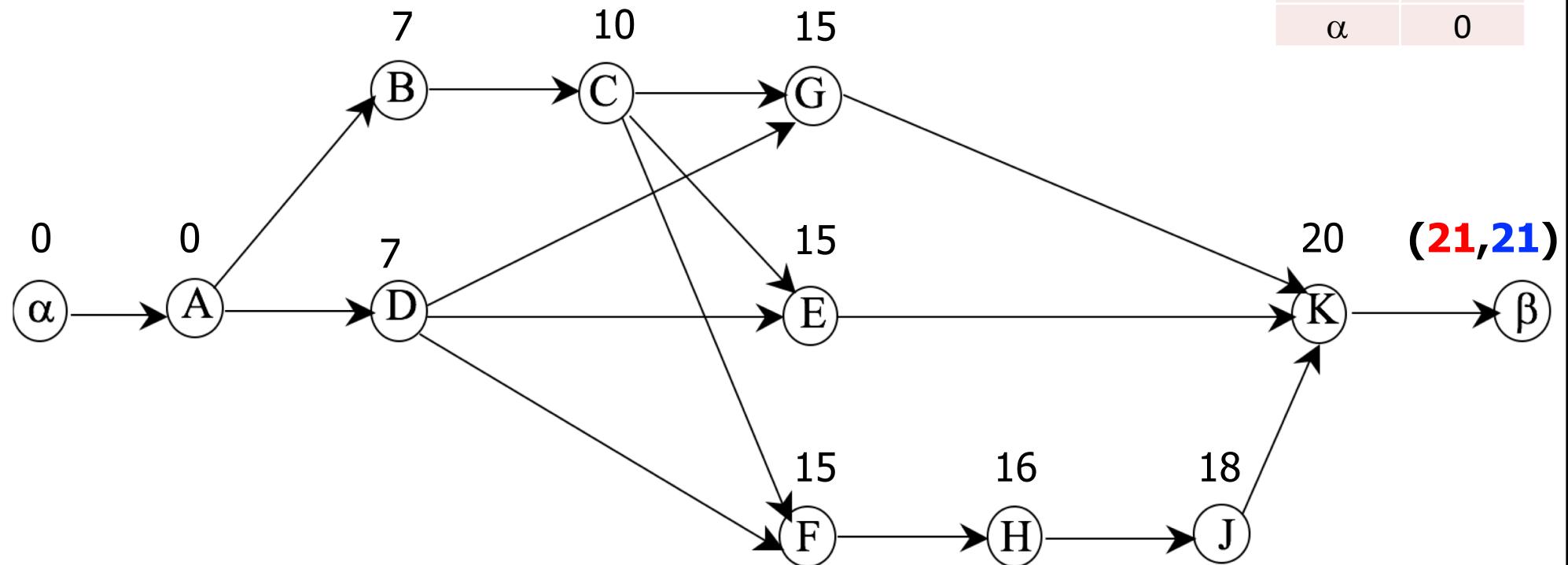
- Tính $T[u]$
 - Xét các đỉnh u theo thứ tự hạng **giảm dần**
 - $T[u] = \min \{T[v]\} - d[u]$ với v là đỉnh kề của u



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

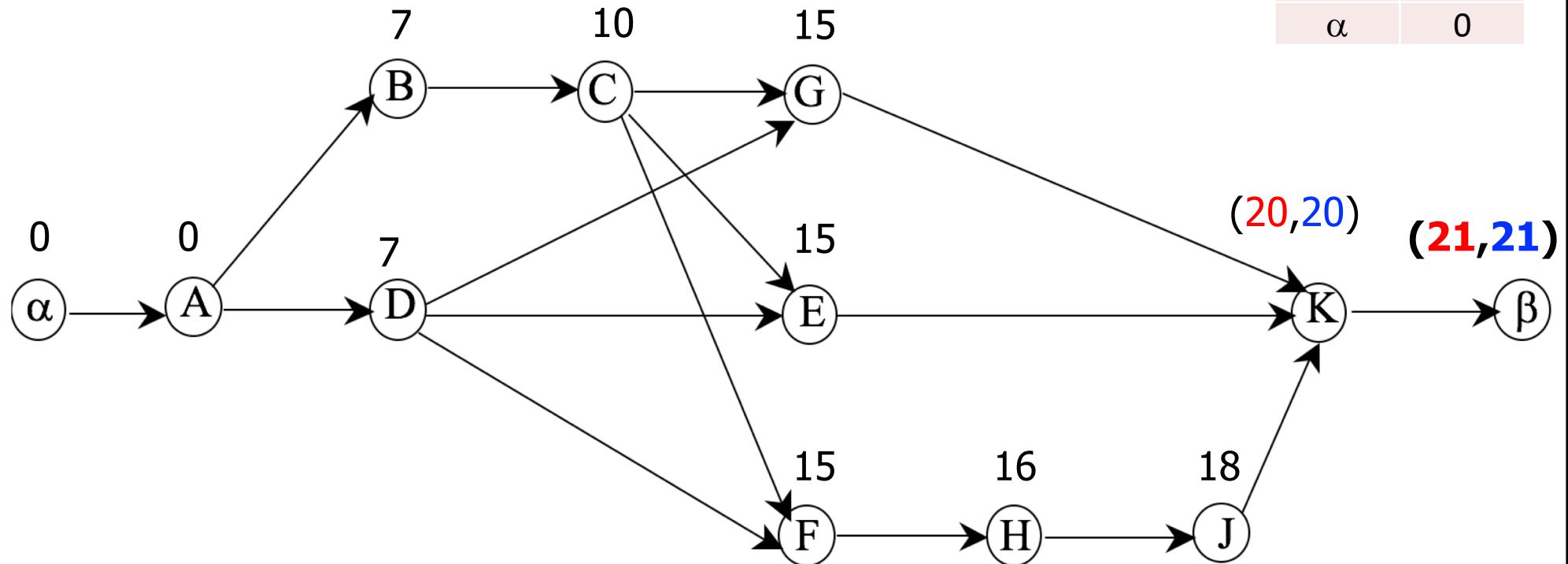
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

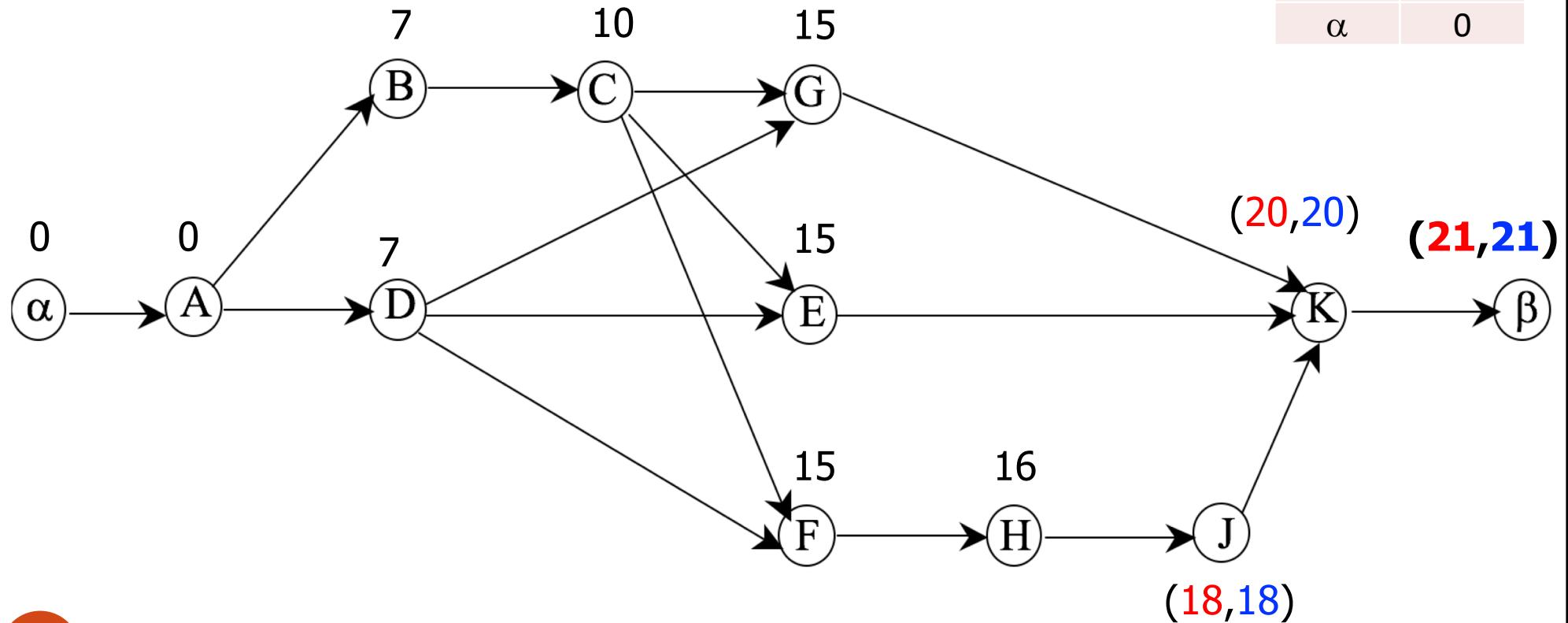
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

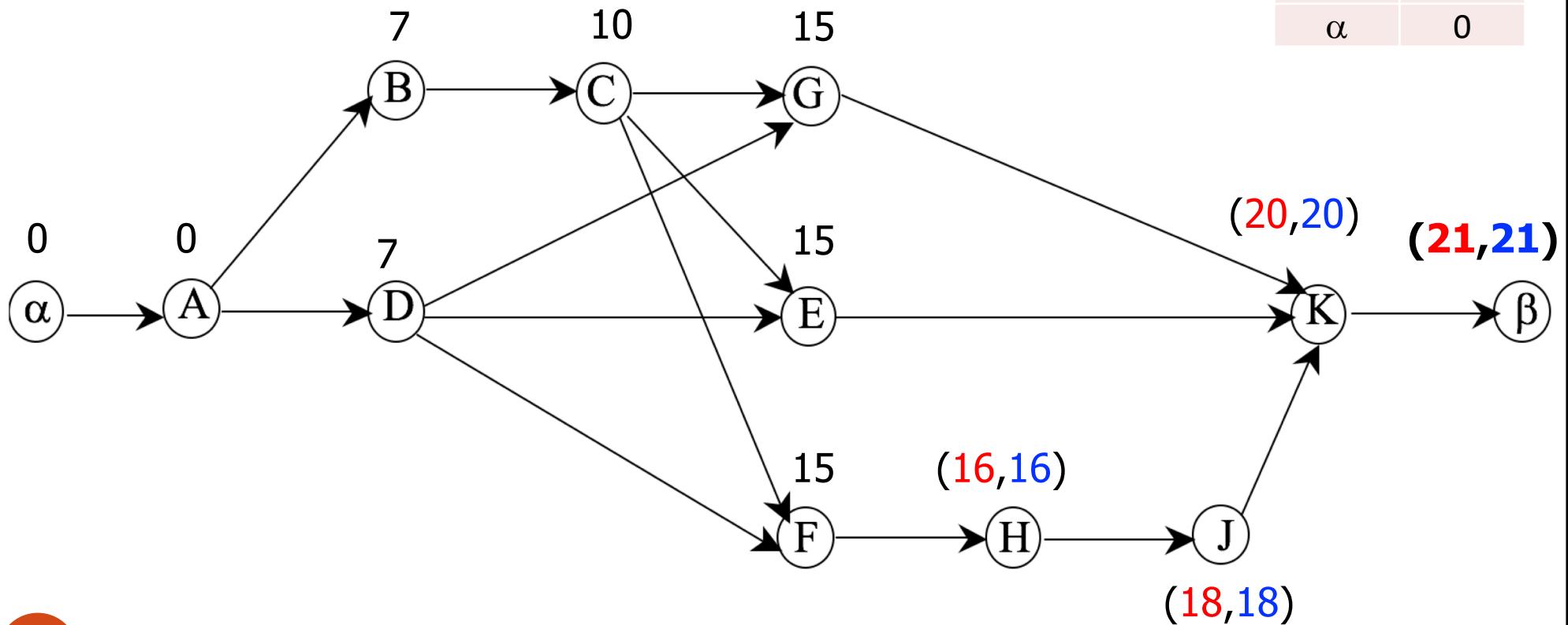
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



Quản lý dự án

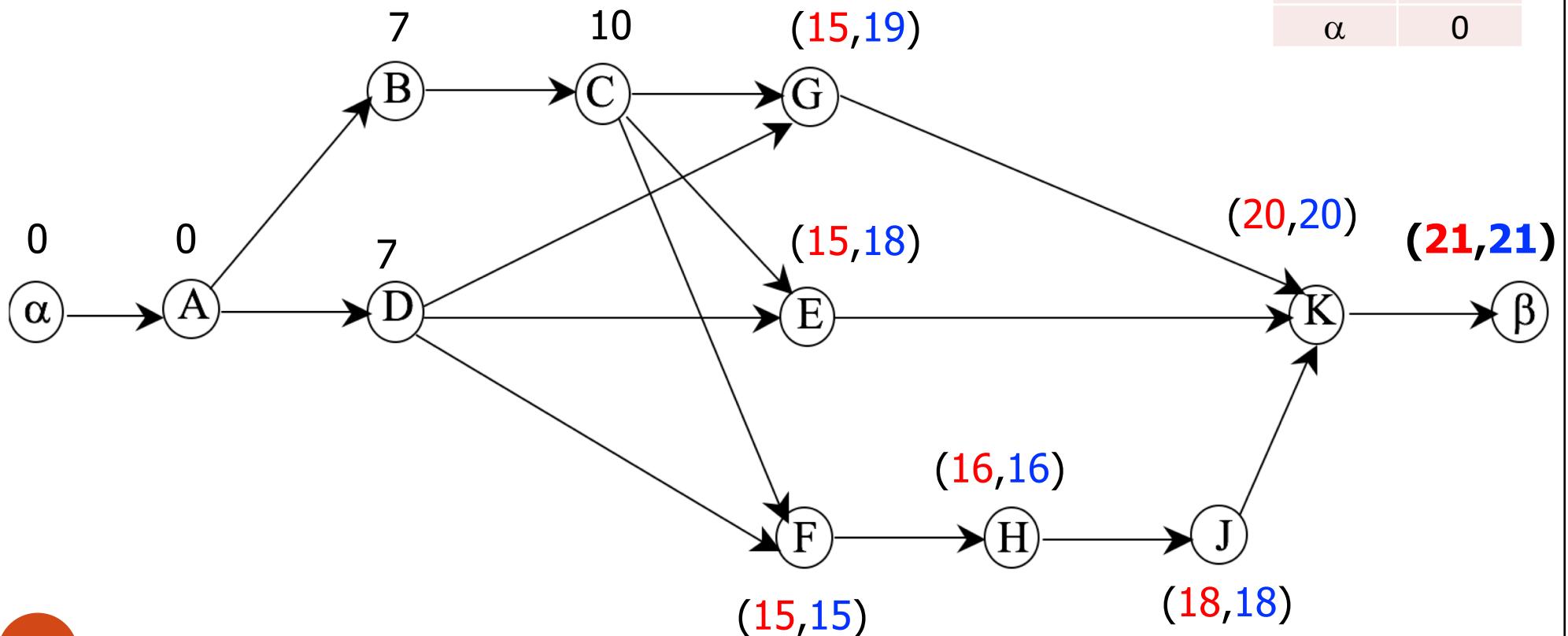
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$

A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0



Quản lý dự án

- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$

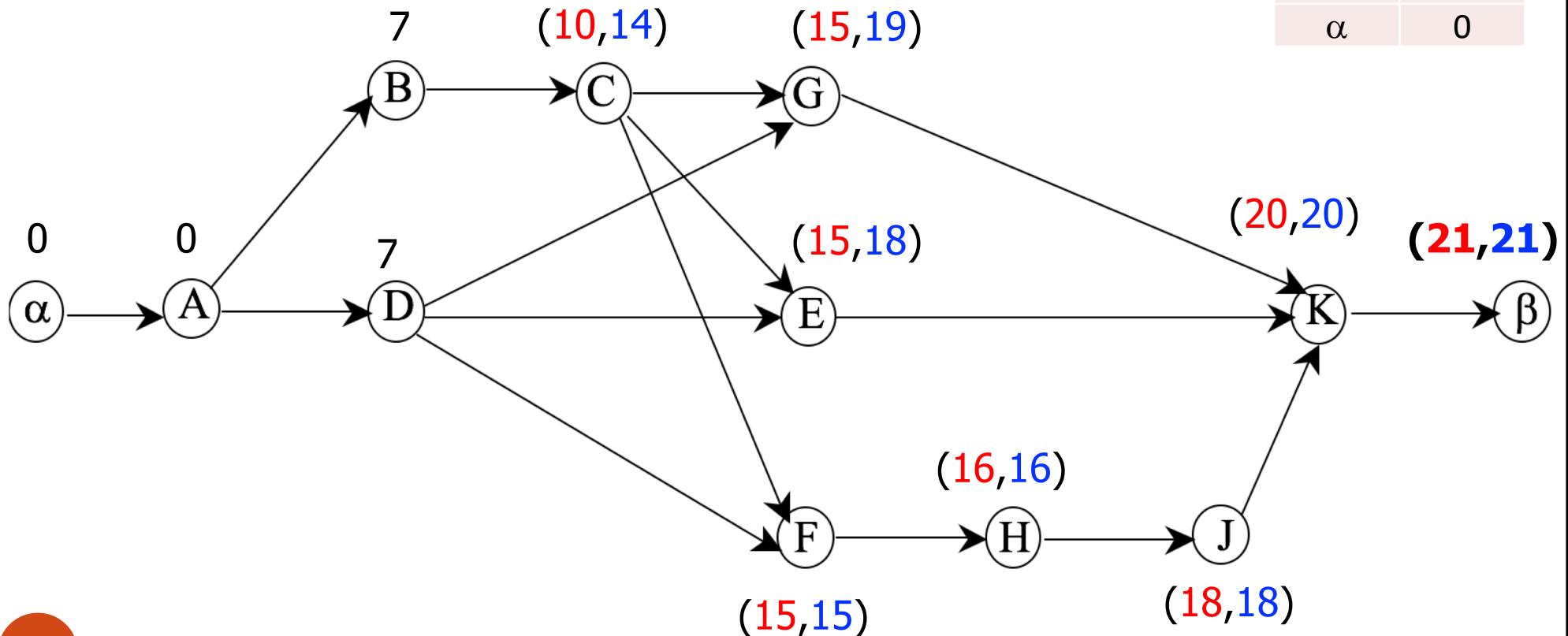


A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

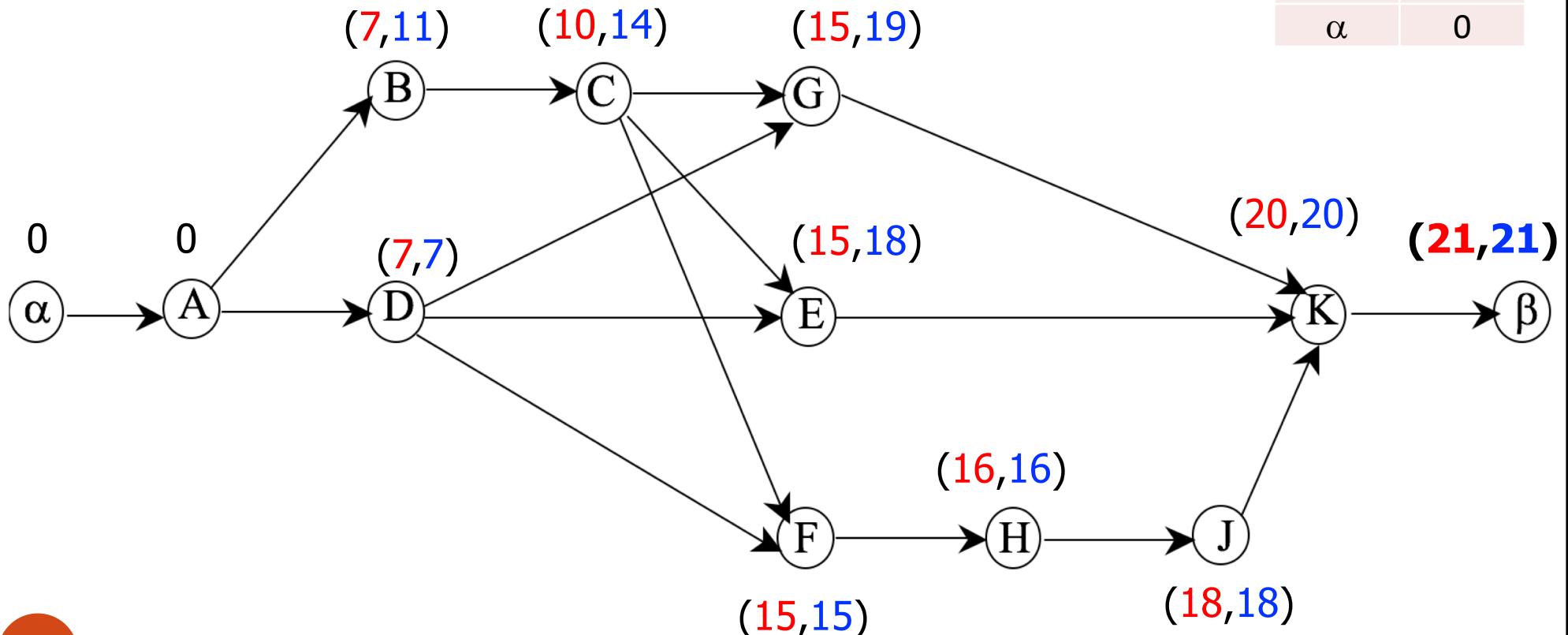
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

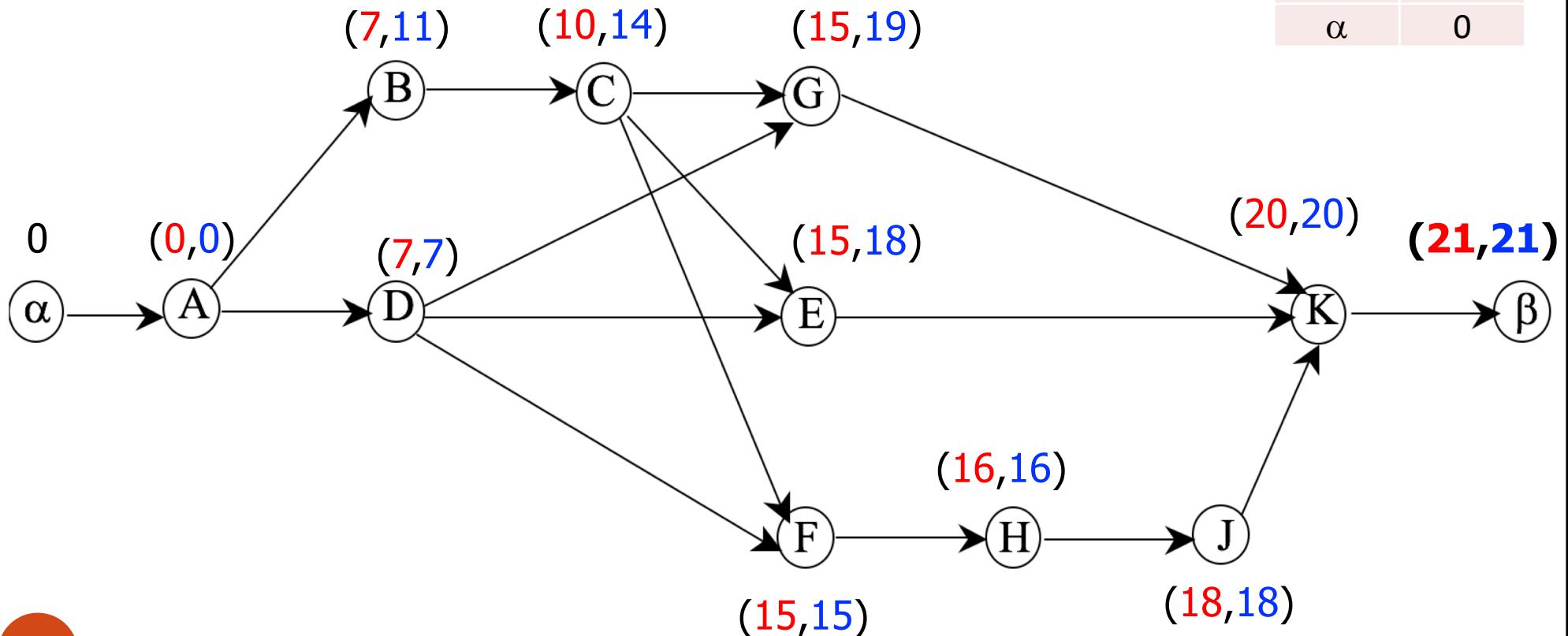
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

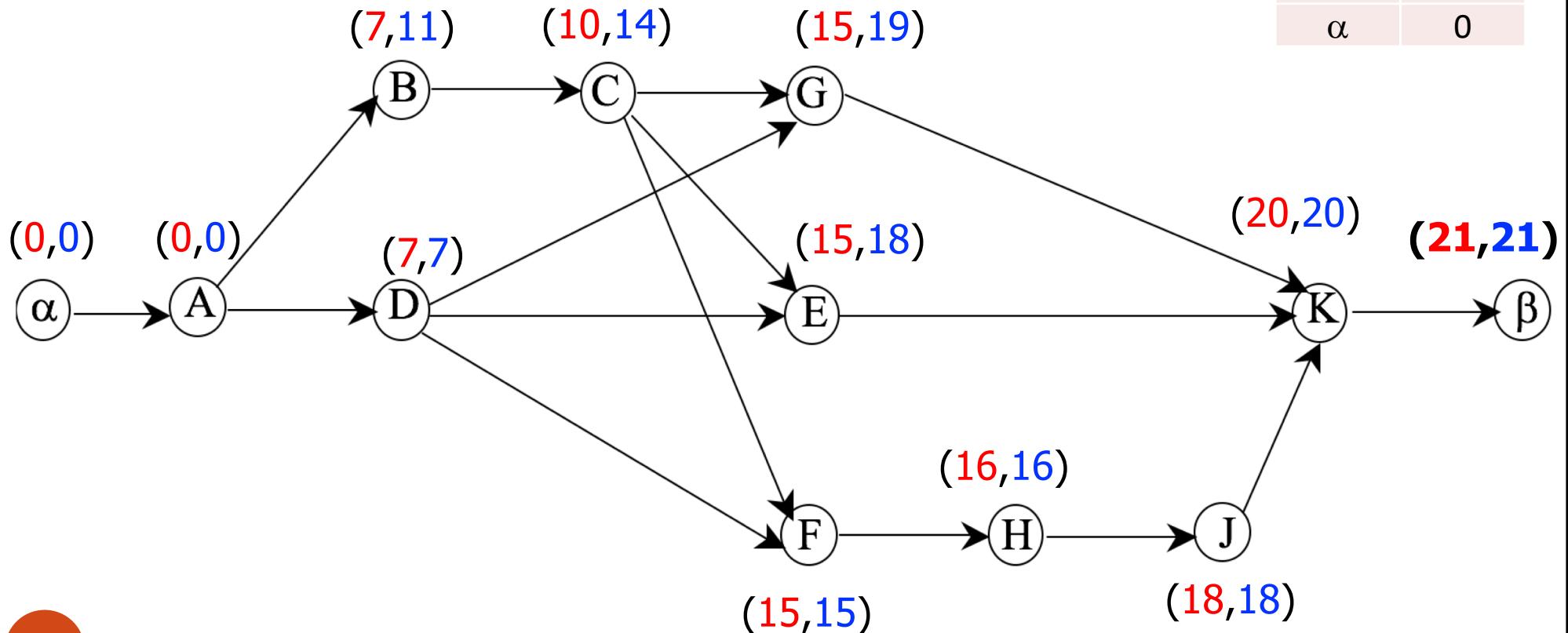
- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

- Áp dụng tìm $t[u]$ và $T[u]$
 - Tìm $T[u]$



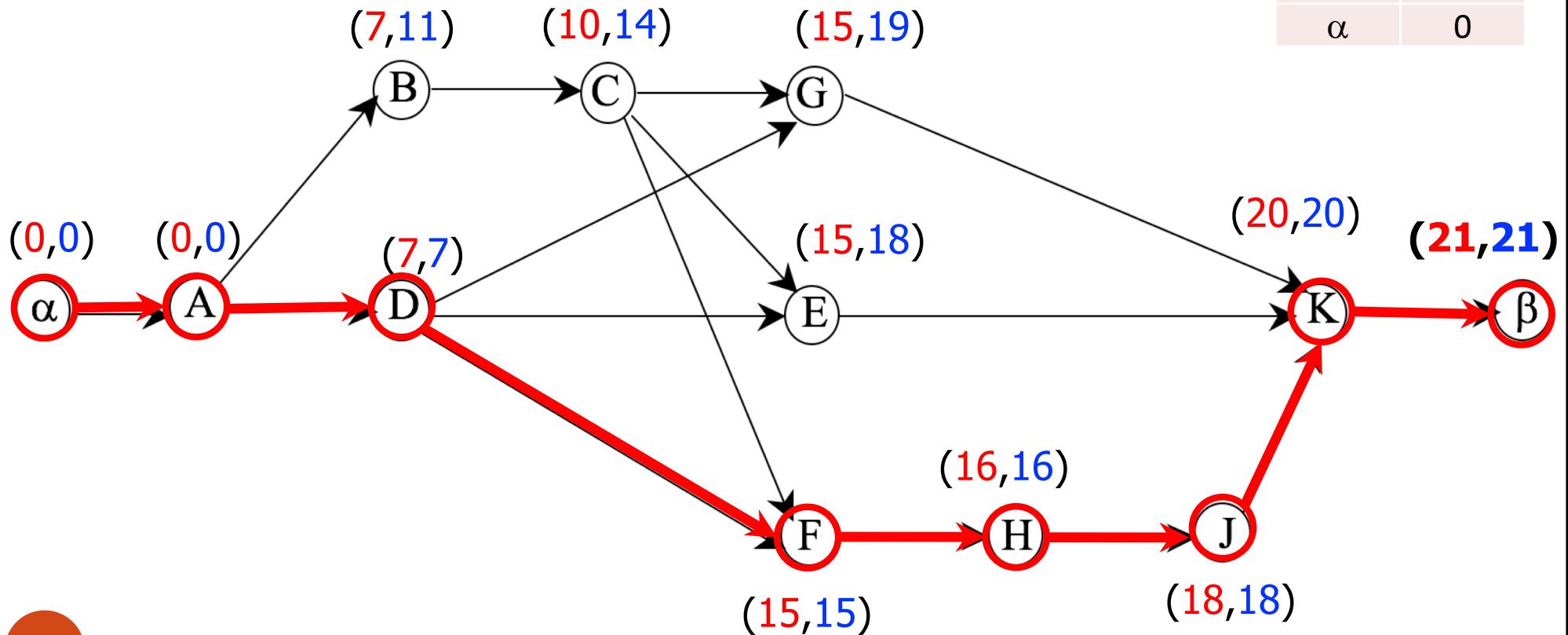
Quản lý dự án

- Công việc then chốt (critical activities)
 - $t[u] = T[u]$
- Đường nối các công việc then chốt: *đường then chốt* (critical path)

A	7
B	3
C	1
D	8
E	2
F	1
G	1
H	2
J	2
K	1
α	0

Quản lý dự án

- Công việc then chốt & đường then chốt



Cho các bảng công việc của một dự án như bên dưới.

Hãy vẽ đồ thị mô hình hóa bài toán quản lý dự án này. Vẽ lại đồ thị sau khi đã xếp hạng.

Tính thời gian sớm nhất có thể bắt đầu công việc u: $t[u]$

Tính thời gian trễ nhất có thể bắt đầu công việc u: $T[u]$

Quy ước

- Hai công việc giả là α và β . Để gõ được ký hiệu này, gõ $\backslash\alpha$ và $\backslash\beta$ vào định tương ứng.

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

Bảng công việc

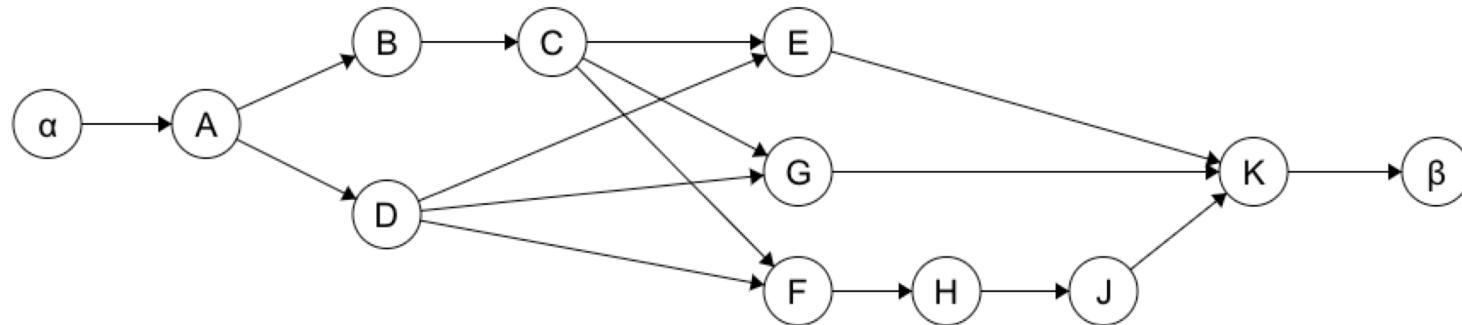
Công việc	Mô tả	Thời gian hoàn thành	Công việc trước đó
A	Các công việc hồ	7	
B	Dựng khung cho mái	3	A
C	Lớp mái	1	B
D	Lắp đặt hệ thống vệ sinh, chiếu sáng	8	A
E	Trang trí mặt tiền	2	C, D
F	Ráp cửa sổ	1	C, D
G	Trang hoàng vườn	1	C, D
H	Làm trần	2	F
J	Sơn phết	2	H
K	Chuyển nhà	1	E, G, J

1. Vẽ đồ thị mô hình hóa bài toán (sắp xếp các đỉnh sao cho đúng với hạng của chúng).

Đồ thị của bài toán



Help Clear shift Delete Edit Undo



2. Tính $t[u]$ và $T[u]$

Đánh dấu * vào ô tương ứng với công việc then chốt

	α	A	B	C	D	E	F	G	H	J	K	β
$d[u]$	0	7	3	1	8	2	1	1	2	2	1	0
$t[u]$	0	0	7	10	7	15	15	15	16	18	20	21
$T[u]$	0	0	11	14	7	18	15	19	16	18	20	21
CV then chốt	*	*			*		*		*	*	*	*