

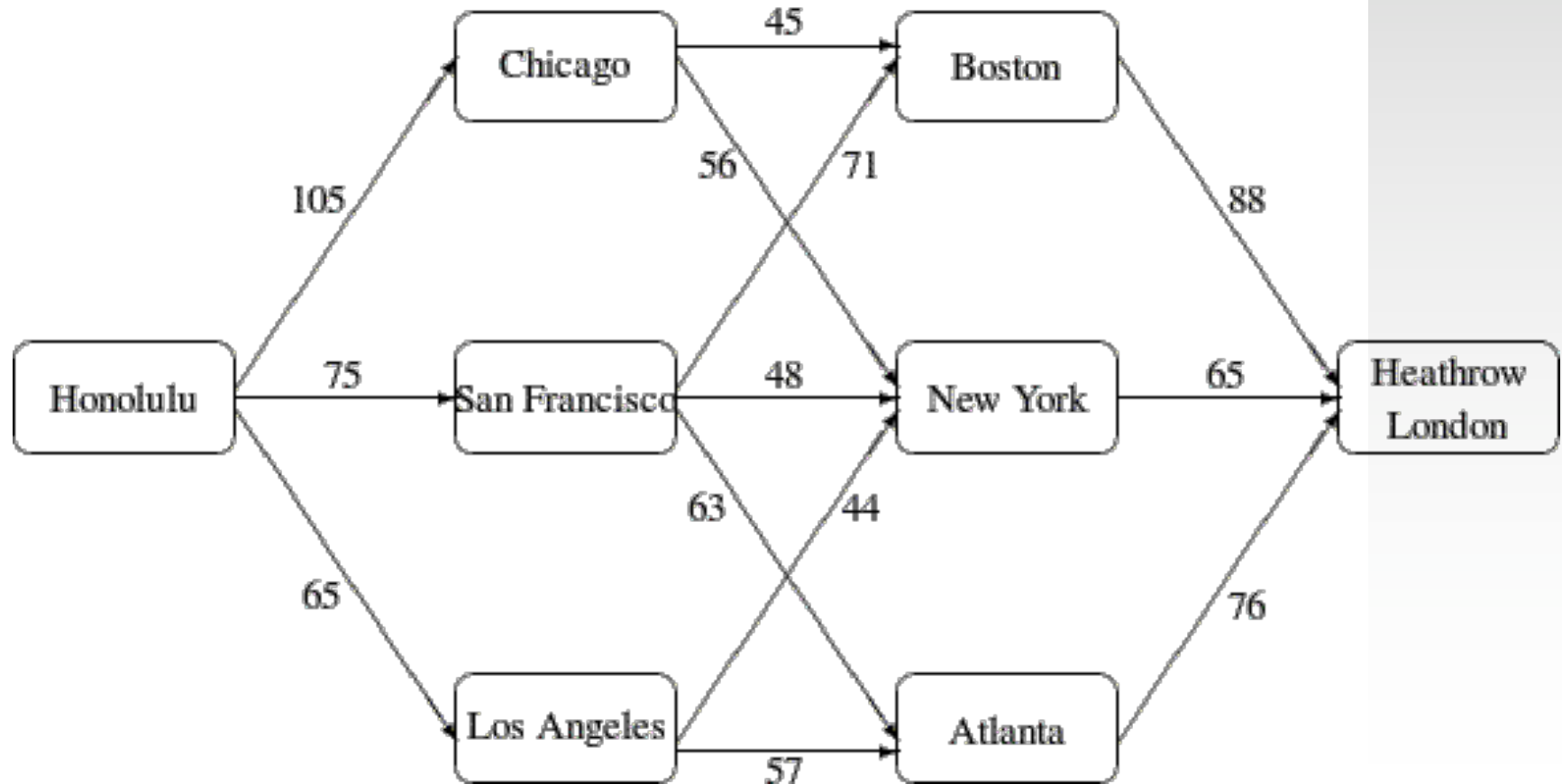
## CHƯƠNG 6

# BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT TRÊN ĐỒ THỊ

# Nội dung

- 1 Bài toán đường đi ngắn nhất
- 2 Thuật toán Dijkstra
- 3 Thuật toán Bellman-Ford
- 4 Thảo luận & Bài tập

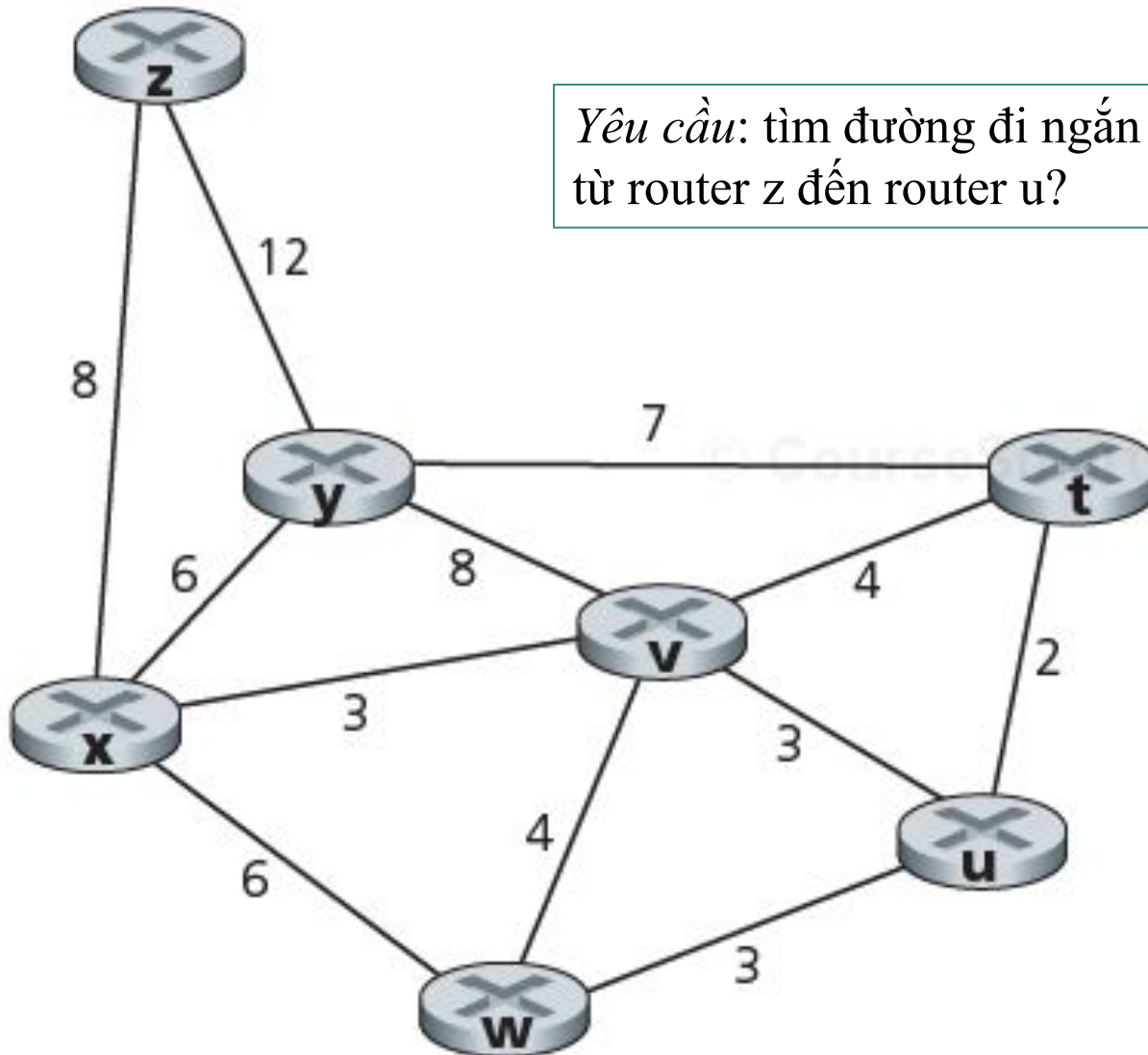
# Bài toán đường đi ngắn nhất (1/5) (SPP – Shortest path problem)



*Yêu cầu:* tìm đường đi ngắn nhất từ Chicago tới Atlanta?

## Bài toán đường đi ngắn nhất (2/5)

*Yêu cầu: tìm đường đi ngắn nhất để truyền gói tin từ router z đến router u?*



# Bài toán đường đi ngắn nhất (3/5)

## Phát biểu

- Cho đồ thị có hướng, có trọng số  $G(V,E,C)$

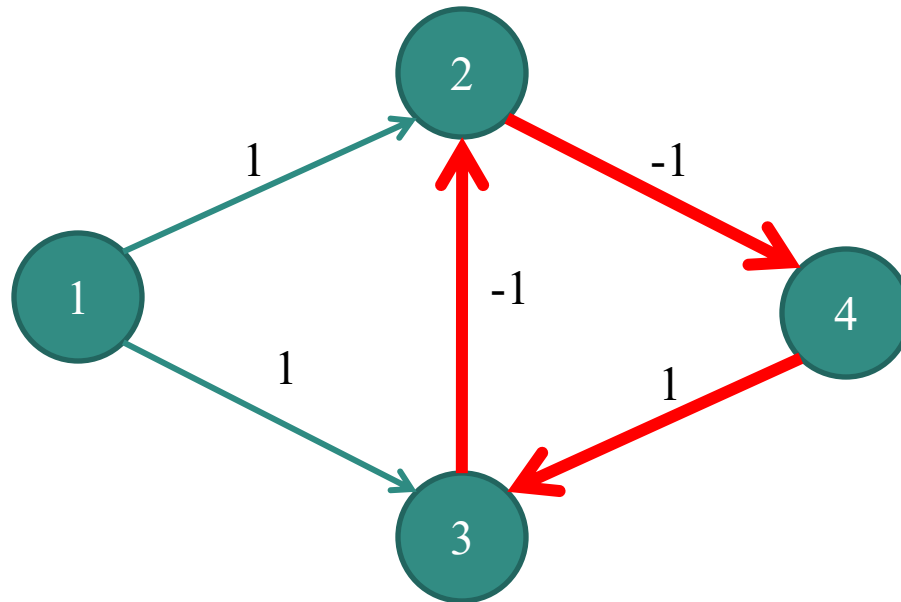
## Yêu cầu

- Tìm đường đi ngắn nhất giữa 2 đỉnh bất kỳ
- *Tìm đường đi ngắn nhất từ một đỉnh đến mọi đỉnh còn lại của đồ thị*
- Tìm đường đi ngắn nhất giữa mọi cặp đỉnh của đồ thị

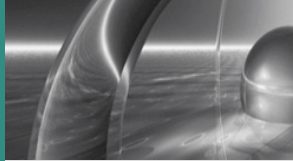
# Bài toán đường đi ngắn nhất (4/5)

## Điều kiện tồn tại đường đi ngắn nhất

- Đồ thị không chứa chu trình âm
- Chu trình âm là chu trình có tổng trọng số âm.



# Bài toán đường đi ngắn nhất (5/5)



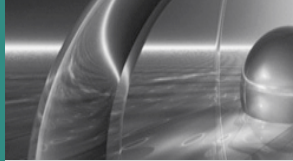
## ❖ Đầu vào:

- Đồ thị có hướng, có trọng số  $G(V,E,C)$ .
- $[G$  không chứa chu trình âm].
- Đỉnh xuất phát  $s$ .

## ❖ Đầu ra:

- Đường đi ngắn nhất từ  $s$  đến các đỉnh còn lại của  $G$ .
- Đường đi ngắn nhất là đường đi có tổng trọng số nhỏ nhất.

# Thuật toán Dijkstra (1/12)



## ❖ Đặc điểm:

- Chỉ áp dụng với đồ thị có trọng số **không âm**.
- Tìm đường đi ngắn nhất từ 1 đỉnh xuất phát đến mọi đỉnh còn lại của đồ thị.

## ❖ Ý tưởng: sử dụng nguyên lý tham lam (greedy)

- Tại mỗi bước luôn chọn đường đi ngắn nhất có thể.
- Đường đi ngắn nhất tới đỉnh chưa xét được xây dựng từ đường đi ngắn nhất qua các đỉnh đã được xét.



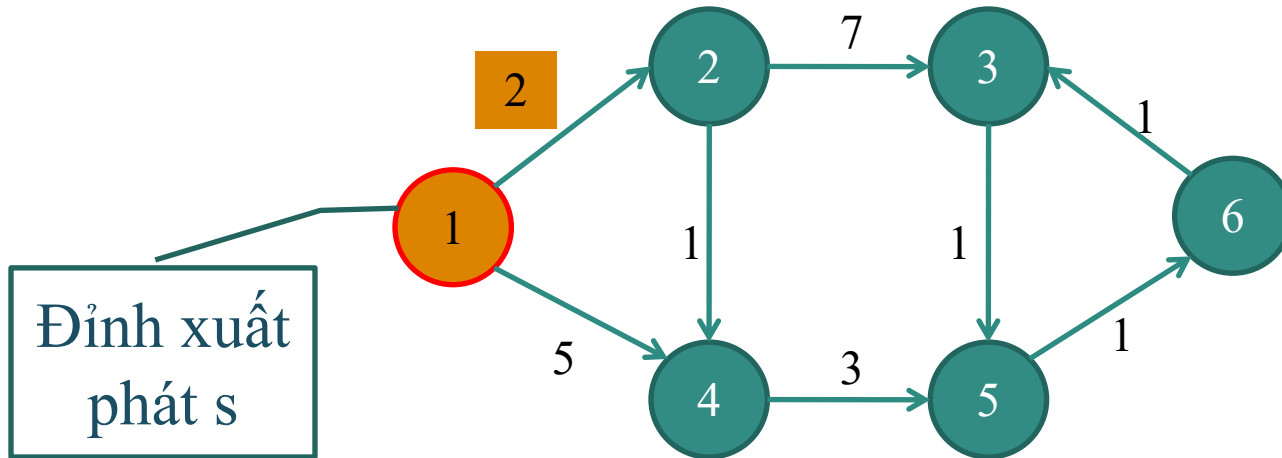
# Thuật toán Dijkstra (2/12)

Dijkstra(V,E,C,s){

- Với mỗi đỉnh  $v$  thuộc  $V$  gán:
  - $\text{distance}(v) = \text{vô cùng}$ ;  $\text{previous}(v) = s$ ;
- $\text{Distance}(s)=0$ ;  $\text{Queue} = V \setminus \{s\}$ ;  $\text{SP} = s$ ;
- While(Queue khác rỗng){
  - **Lấy  $x$  là đỉnh có  $\text{distance}(x) = \text{Min } \{\text{distance}(\text{Queue})\}$**
  - Nạp  $x$  vào SP
  - Với mỗi đỉnh  $y$  thuộc Queue và kề với  $x$ ,
  - nếu  $\text{distance}(y) > \text{distance}(x) + c(x,y)$  {
    - Cập nhật  $\text{distance}(y) = \text{distance}(x) + c(x,y)$ ;
    - Cập nhật  $\text{previous}(y) = x$ ;
  - }
- }
- Return SP;

}

# Thuật toán Dijkstra (3/12)

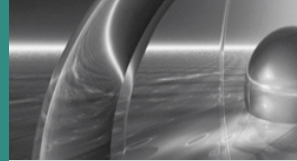


Đỉnh =>	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1

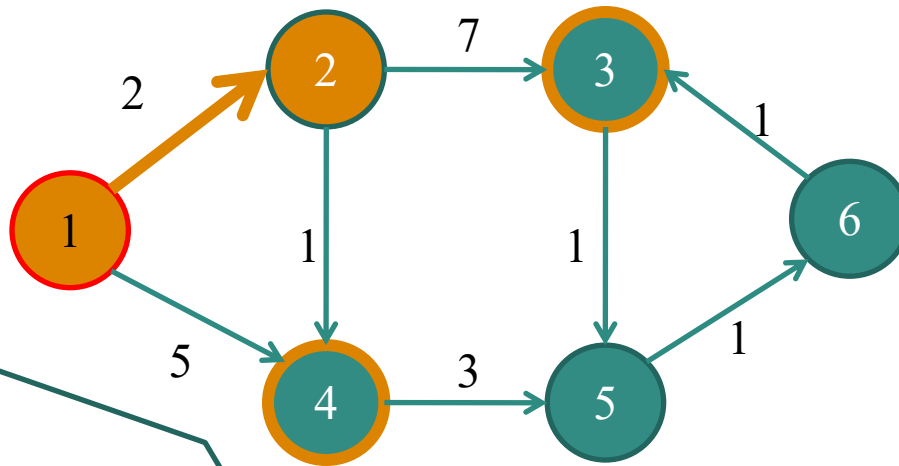
Distance(2)=2

Previous(2)=1

# Thuật toán Dijkstra (4/12)



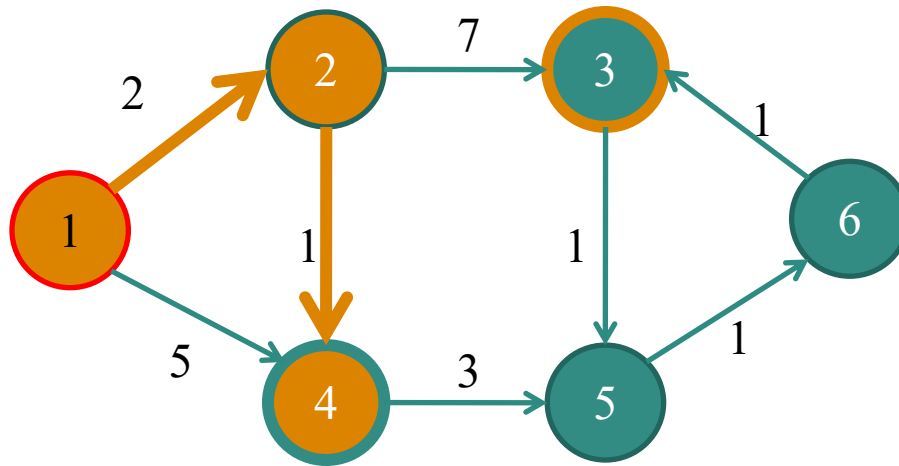
Chọn đỉnh  
có distance  
nhỏ nhất



	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,2	---	---

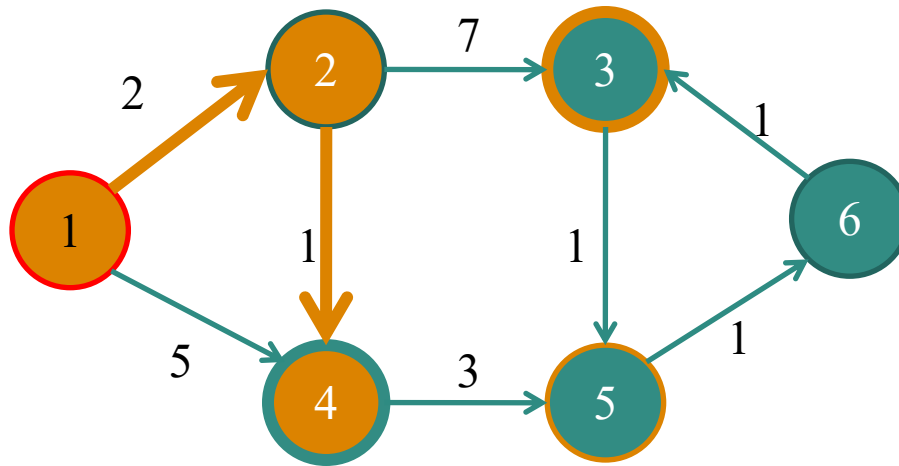
Cập nhật lại distance và  
previous của các đỉnh kề với  
đỉnh mới được chọn

# Thuật toán Dijkstra (5/12)



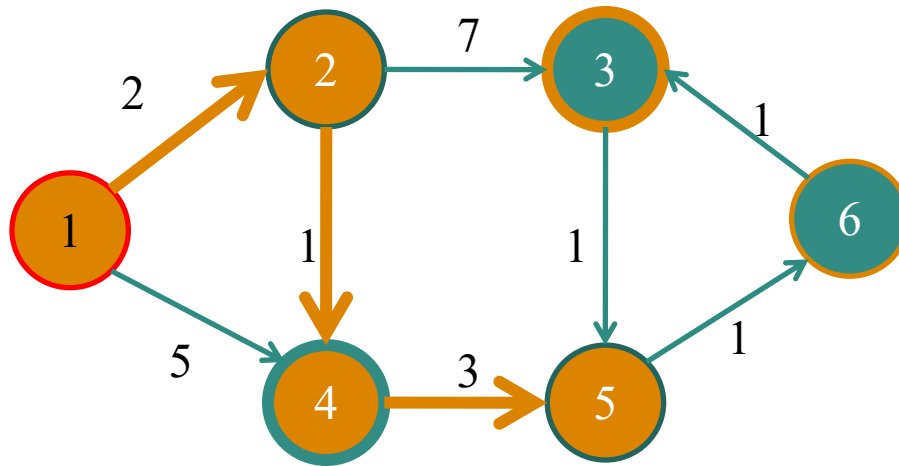
	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,2	---	---

# Thuật toán Dijkstra (6/12)



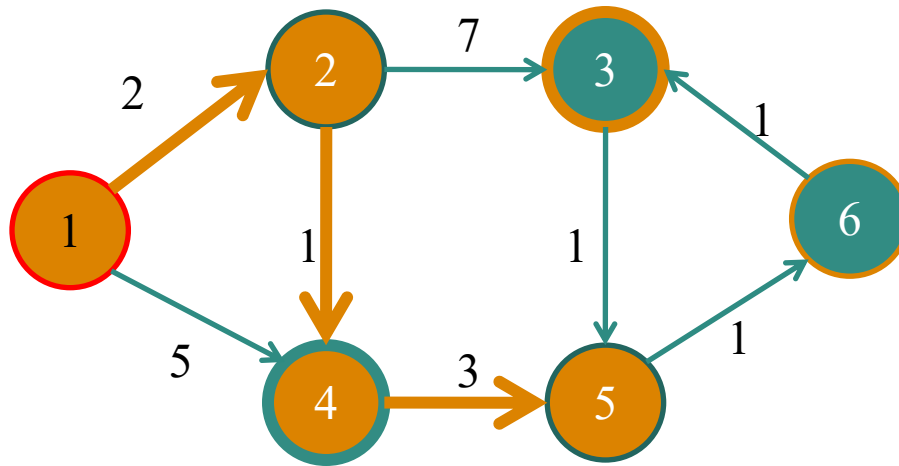
	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,2	---	---
2			9,2		6,4	---

# Thuật toán Dijkstra (7/12)



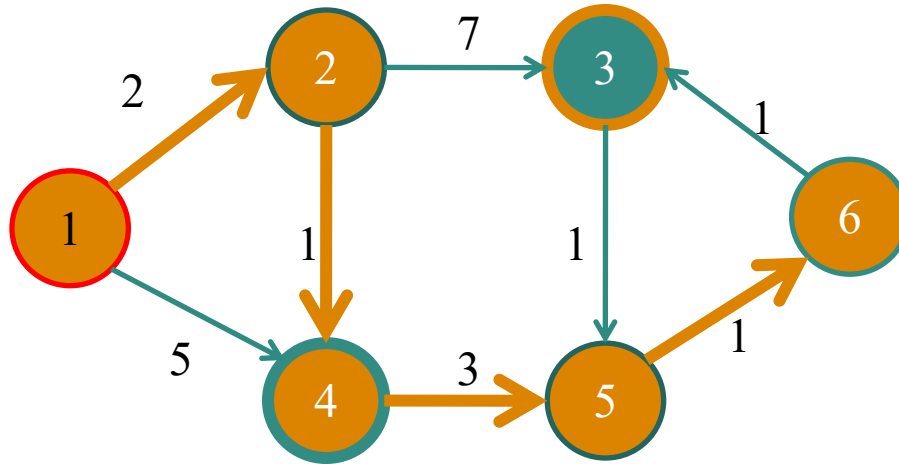
	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,2	---	---
2			9,2		6,4	---

# Thuật toán Dijkstra (8/12)



	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,2	---	---
2			9,2		6,4	---
3			9,2			7,5

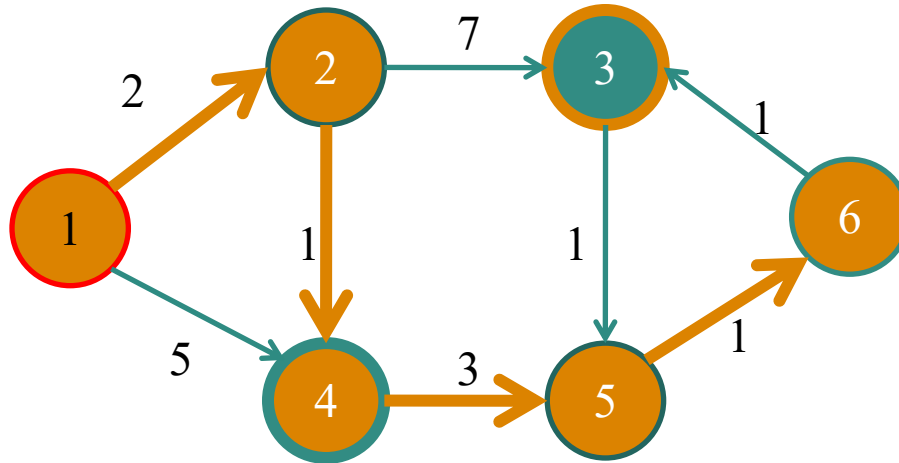
# Thuật toán Dijkstra (9/12)



	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,2	---	---
2			9,2		6,4	---
3			9,2			7,5

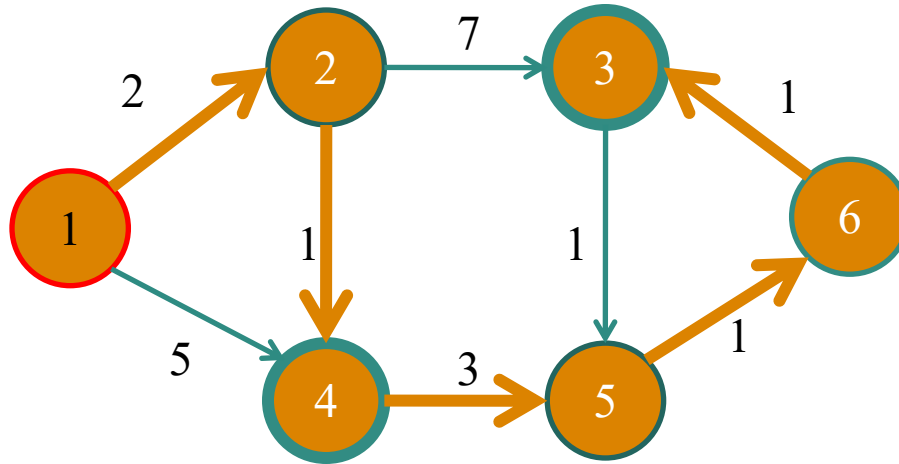


# Thuật toán Dijkstra (10/12)



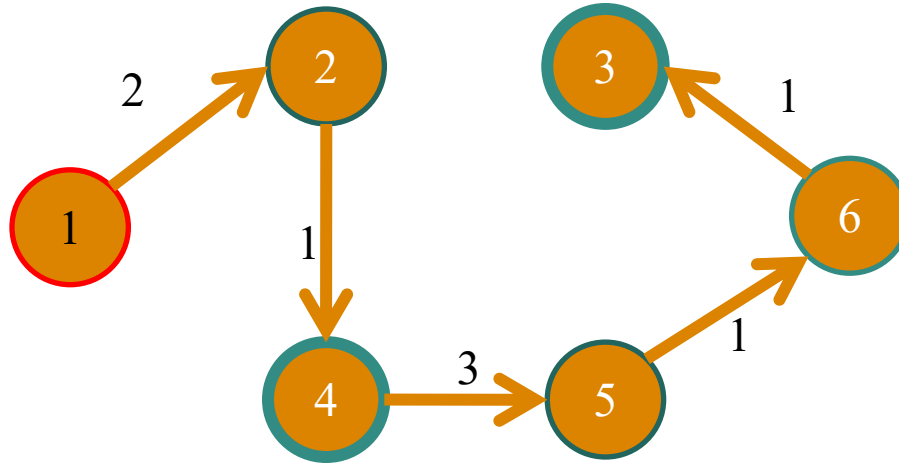
	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,1	---	---
2			9,2		6,4	---
3			9,2			7,5
			8,6			

# Thuật toán Dijkstra (11/12)

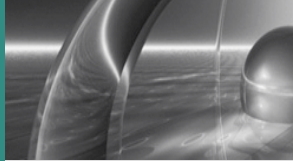


	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,1	---	---
2			9,2		6,4	---
3			9,2			7,5
4			8,6			

# Thuật toán Dijkstra (12/12)



	1	2	3	4	5	6
0	0,1	2,1	$\infty$ ,1	5,1	$\infty$ ,1	$\infty$ ,1
1			9,2	3,1	---	---
2			9,2		6,4	---
3			9,2			7,5
4			8,6			
Kết luận	0,1	2,1	8,6	3,2	6,4	7,5



## ❖ Đầu vào:

- Đồ thị có hướng, có trọng số  $G(V, E, C, s)$ .
- Trong đó  $s$  là đỉnh xuất phát,  $C$  là trọng số.

## ❖ Đầu ra:

- Đường đi ngắn nhất từ  $s$  đến mọi đỉnh còn lại.
- Hoặc đồ thị chứa chu trình âm.



# Thuật toán Bellman – Ford (2/11)

## ❖ Khởi tạo

- $\text{Distance}(s) = 0$
- Với mọi  $v$  thuộc  $V \setminus \{s\}$ :  $\text{distance}(v) = \infty$ ,  $\text{previous}(v) = s$

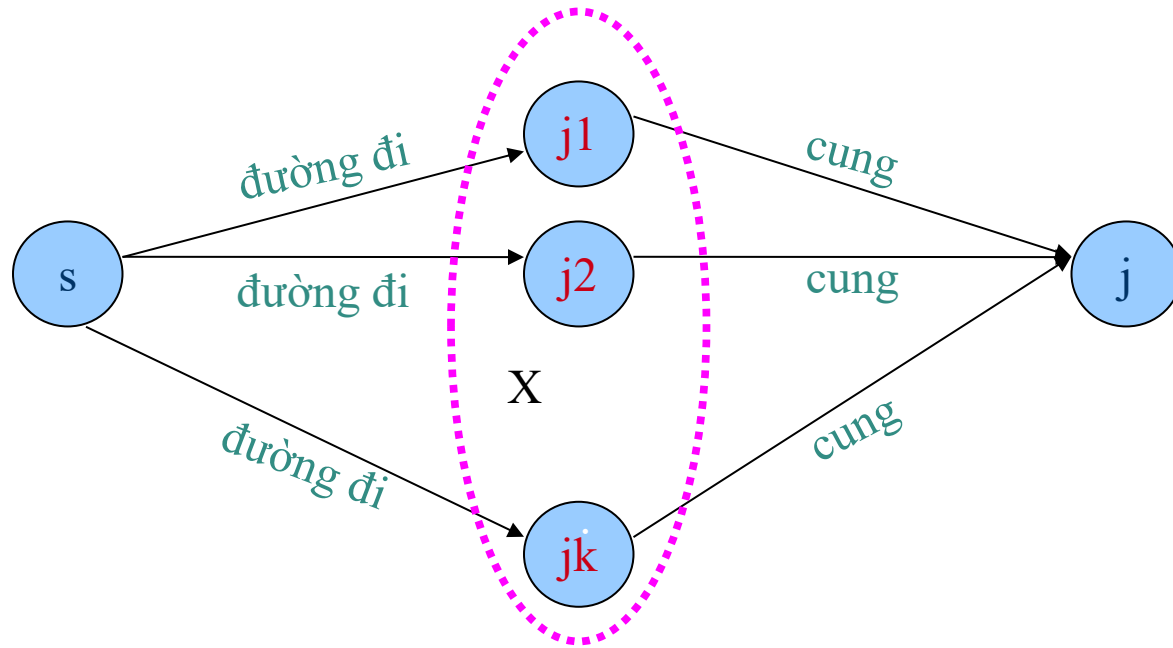
## ❖ Với $k = 1$ đến $|V| - 1$ làm:

- Với mỗi cung  $(u,v)$  thuộc  $E$ :
  - Nếu  $\text{distance}(v) > \text{distance}(u) + c(u,v)$  thì:
    - $\text{distance}(v) = \text{distance}(u) + c(u,v)$
    - Cập nhật  $\text{previous}(v) = u$

## ❖ Với mỗi cạnh $(u,v)$ thuộc $E$ :

- Nếu  $\text{distance}(v) > \text{distance}(u) + c(u,v)$  thì đồ thị chứa chu trình âm.

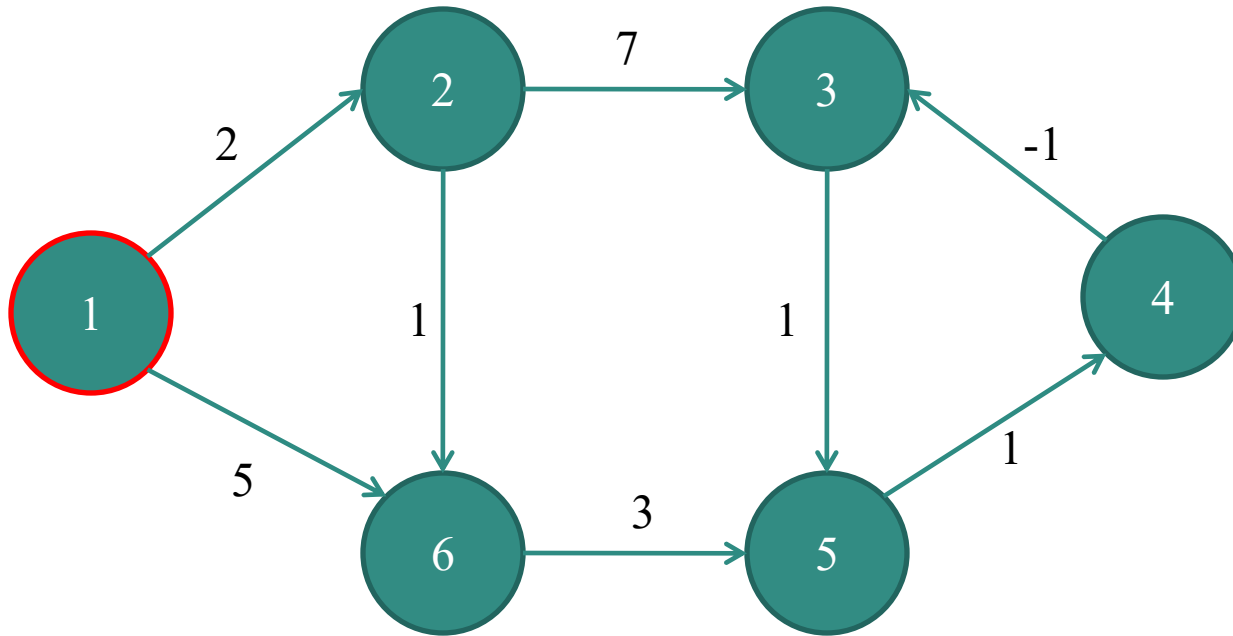
# Thuật toán Bellman – Ford (3/11)



**Nguyên lý Bellman**

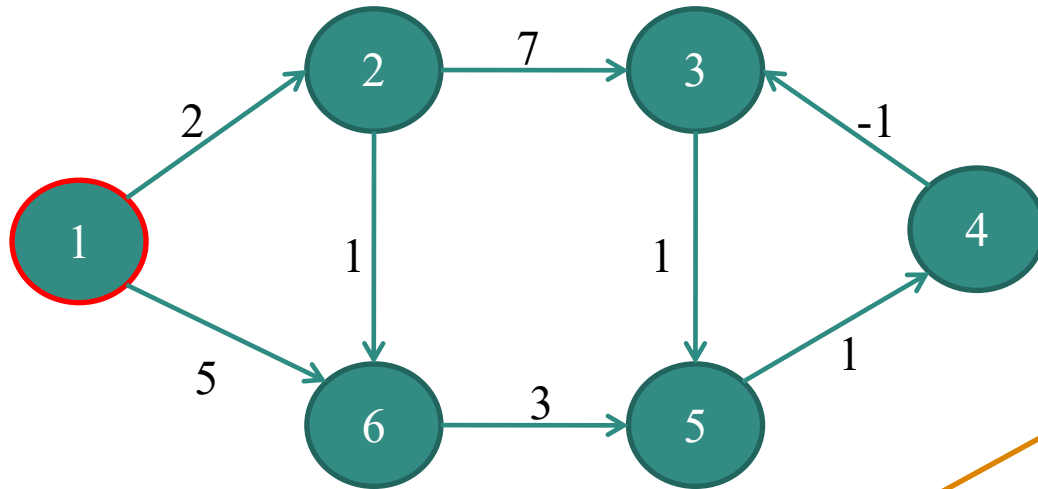
$$\text{distance}(j) = \underset{j_k \in X}{\text{Min}} \{ \text{distance}(j_k) + c(j_k, j) \}$$

# Thuật toán Bellman – Ford (4/11)



Đỉnh =>	1	2	3	4	5	6
0	0,1	$\infty$ ,1	$\infty$ ,1,	$\infty$ ,1	$\infty$ ,1	$\infty$ ,1

# Thuật toán Bellman – Ford (5/11)



Xét  $j = 2$ :

$X = \{1\}$

$\text{Distance}(j) = \text{Min}\{\infty, \textcolor{red}{0} + 2\} = 2$

$\text{Previous}(j) = 1$

Đỉnh =>	1	2	3	4	5	6
<b>0</b>	0,1	<b><math>\infty</math></b> ,1	$\infty$ ,1,	$\infty$ ,1	$\infty$ ,1	$\infty$ ,1
1		2,1				

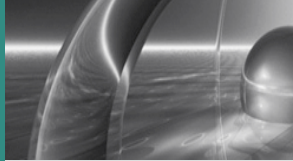
Xét  $j = 3$ :

$X = \{2\}$

$\text{Distance}(3) = \text{Min}\{\textcolor{red}{\infty}, \infty\} = \infty$

=> Không cập nhật  $\text{distance}(3)$  và  $\text{Previous}(3)$





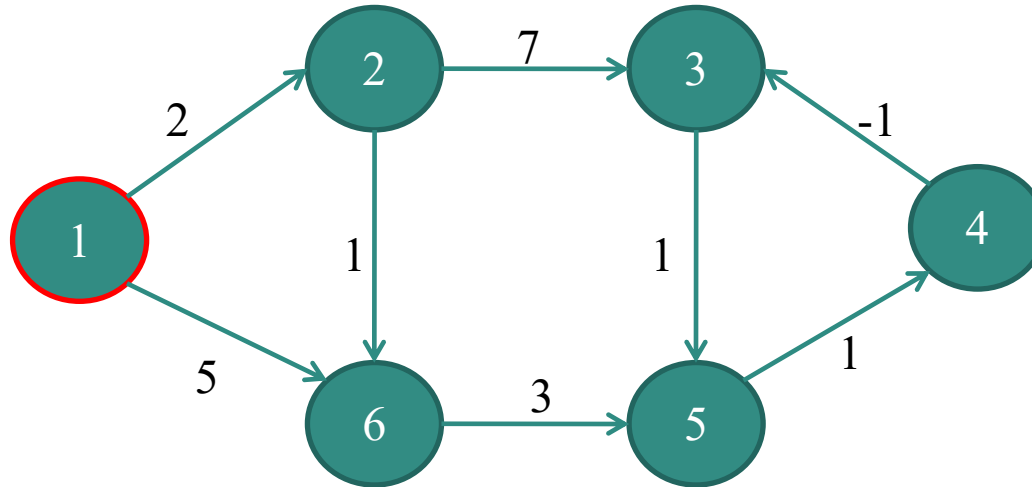
## ❖ Giải thích:

- Sở dĩ chúng ta vẫn chọn  $\text{distance}(3) = \text{Min}\{\infty, \infty\}$  trong đó  $\infty$  là giá trị “cũ” của  $\text{distance}(2)$  [*trong khi giá trị “mới” của  $\text{distance}(2)$  là 2*] vì trong trường hợp tổng quát đỉnh  $j$  có thể được chọn tùy ý – tức là đỉnh 3 có thể được xét trước đỉnh 2!



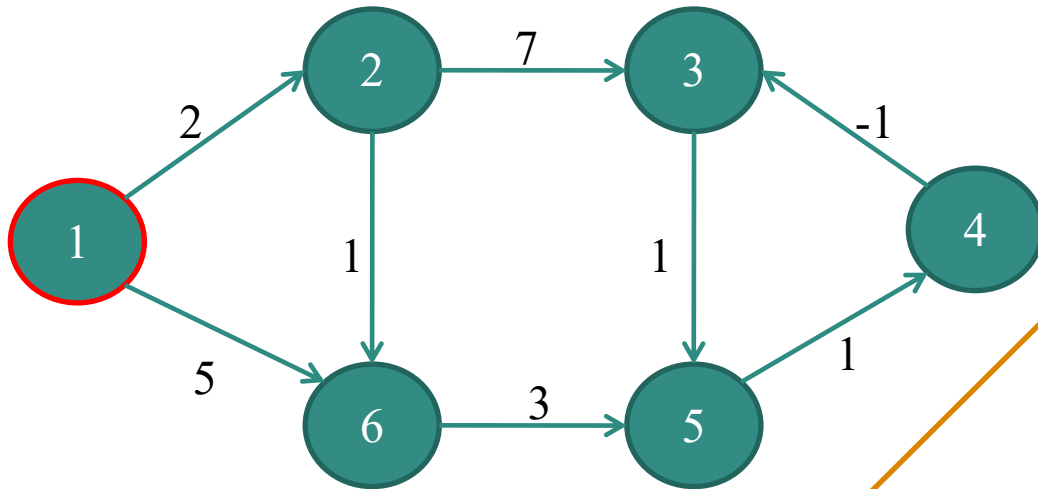
# Thuật toán Bellman – Ford (7/11)

❖ Tương tự với  $j = 4, 5$  và  $6$  ta được:



Đỉnh =>	1	2	3	4	5	6
0	0,1	$\infty$ ,1	$\infty$ ,1,	$\infty$ ,1	$\infty$ ,1	$\infty$ ,1
1		2,1	-----	-----	-----	5,1

# Thuật toán Bellman – Ford (8/11)



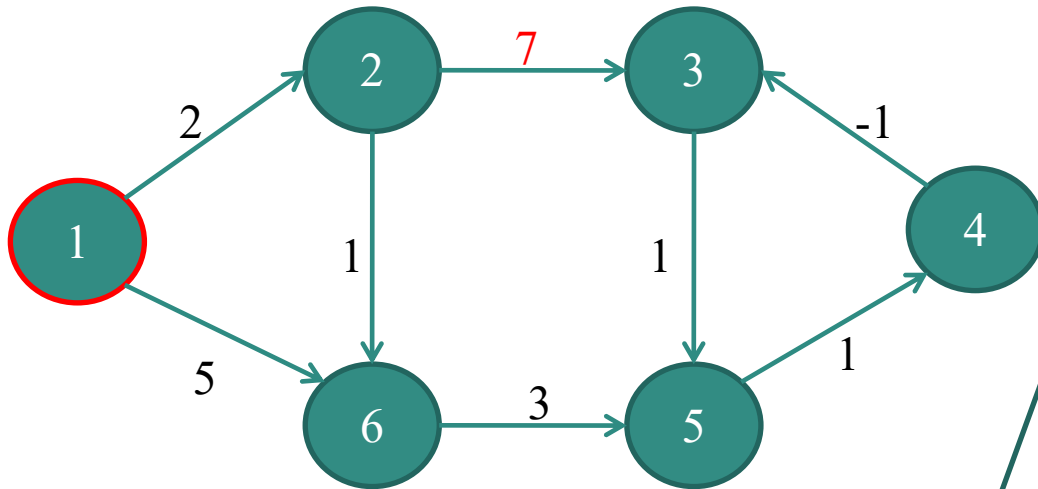
*Quá trình lặp lại:*

Xét  $j = 2$ :

$X = \{1\} \Rightarrow$  Không cập nhật  $\text{distance}(2)$  và  $\text{previous}(2)$

Đỉnh =>	1	2	3	4	5	6
0	0,1	$\infty, 1$	$\infty, 1,$	$\infty, 1$	$\infty, 1$	$\infty, 1$
1		2,1	-----	-----	-----	5,1
2		2,1				

# Thuật toán Bellman – Ford (9/11)



*Quá trình lặp lại:*

Xét  $j = 3$ :

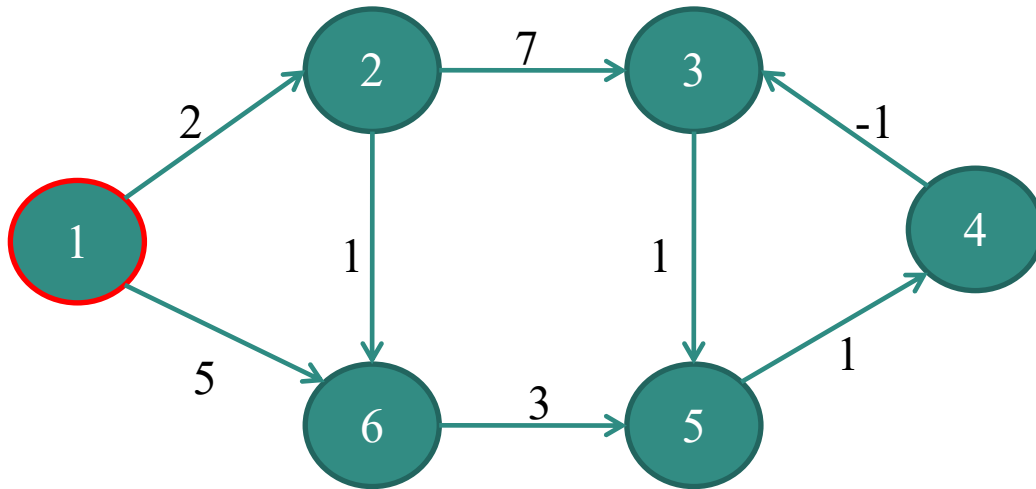
$X = \{2\} \Rightarrow$

$\text{distance}(3) = \min\{\infty, 2+7\}=9$

Cập nhật  $\text{previous}(3) = 2$

Đỉnh =>	1	2	3	4	5	6
0	0,1	$\infty,1$	$\infty,1,$	$\infty,1$	$\infty,1$	$\infty,1$
1		2,1	-----	-----	-----	5,1
2		2,1	9,2			

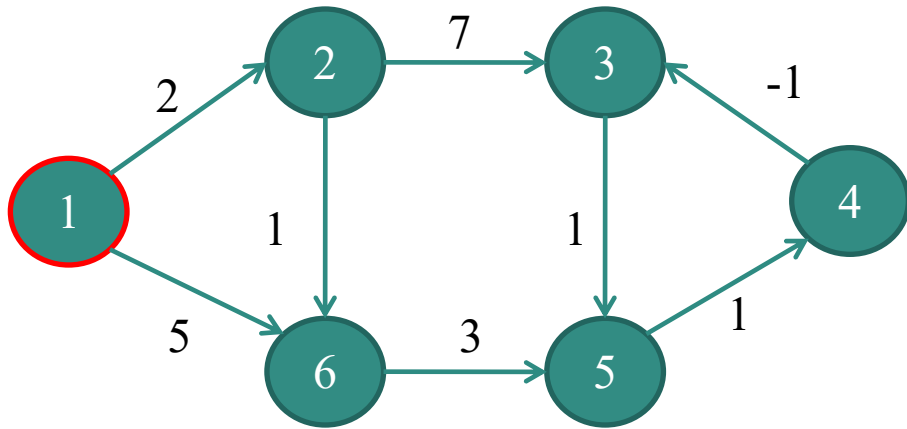
# Thuật toán Bellman – Ford (10/11)



*Tương tự với  $j = 4, 5$  và  $6$ :*

Đỉnh =>	1	2	3	4	5	6
0	0,1	$\infty$ ,1	$\infty$ ,1,	$\infty$ ,1	$\infty$ ,1	$\infty$ ,1
1		2,1	-----	-----	-----	5,1
2		2,1	9,2	-----	8,6	3,2

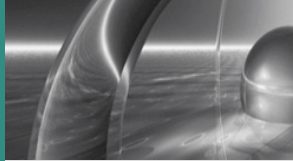
# Thuật toán Bellman – Ford (11/11)



*Quá trình lặp lại cho đến khi:*

- Trong bảng có 2 dòng liên tiếp trùng nhau!
- Hoặc đã đủ  $n$  dòng! (????)

Đỉnh =>	1	2	3	4	5	6
0	0,1	$\infty$ ,1	$\infty$ ,1,	$\infty$ ,1	$\infty$ ,1	$\infty$ ,1
1		2,1	-----	-----	-----	5,1
2		2,1	9,2	-----	8,6	3,2
3		2,1	9,2	9,5	6,6	3,2
4		2,1	8,4	7,5	6,6	3,2
5		2,1	6,4	7,5	6,6	3,2
6		2,1	6,4	7,5	6,6	3,2

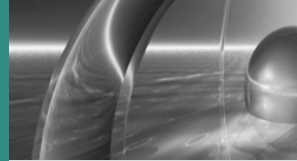


## ❖ Về thuật toán Bellman-Ford:

1. Thuật toán cho kết quả là đường đi ngắn nhất ?
2. Khi nào thuật toán dừng?
3. Thuật toán có thể phát hiện chu trình âm?



# Thảo luận & bài tập (2/3)



## ❖ Giải thích:

- Đường đi ngắn nhất (nếu có) giữa 2 đỉnh bất kỳ chứa nhiều nhất  $n - 1$  cung, với  $n$  là số đỉnh của đồ thị.
- Tại bước lặp thứ  $k$  trong thuật toán, mọi đường đi ngắn nhất qua không quá  $k$  cung được xác định.
- Sau  $|V| - 1$  bước lặp, tất cả các cung  $(u, v)$  đều thỏa điều kiện:  $\text{distance}(v) \leq \text{distance}(u) + c(u, v)$ . Tức là đường đi ngắn nhất từ  $s$  đến  $v$  đã được xác định.
- Nếu (ở bước kế tiếp – bước thứ  $|V|$ ) tồn tại cung  $(u, v)$  nào đó có  $\text{distance}(v) > \text{distance}(u) + c(u, v)$  tức là đồ thị có chu trình âm.
- Trong thực tế, có thể dừng thuật toán ngay khi xuất hiện 2 dòng trùng nhau trong bảng biểu diễn các giá trị distance.



# Thảo luận & bài tập (3/3)

## ❖ Các vấn đề khác:

Vấn đề	Dijkstra	Bellman-Ford
Độ phức tạp tính toán?	?	?
Biểu diễn đồ thị bằng phương pháp nào sẽ phù hợp với thuật toán?		
Giải quyết bài toán trên đồ thị vô hướng?		
Tìm đường đi ngắn nhất đi qua (các) cạnh / cung cho trước?		
Cài đặt thuật toán trên máy tính		