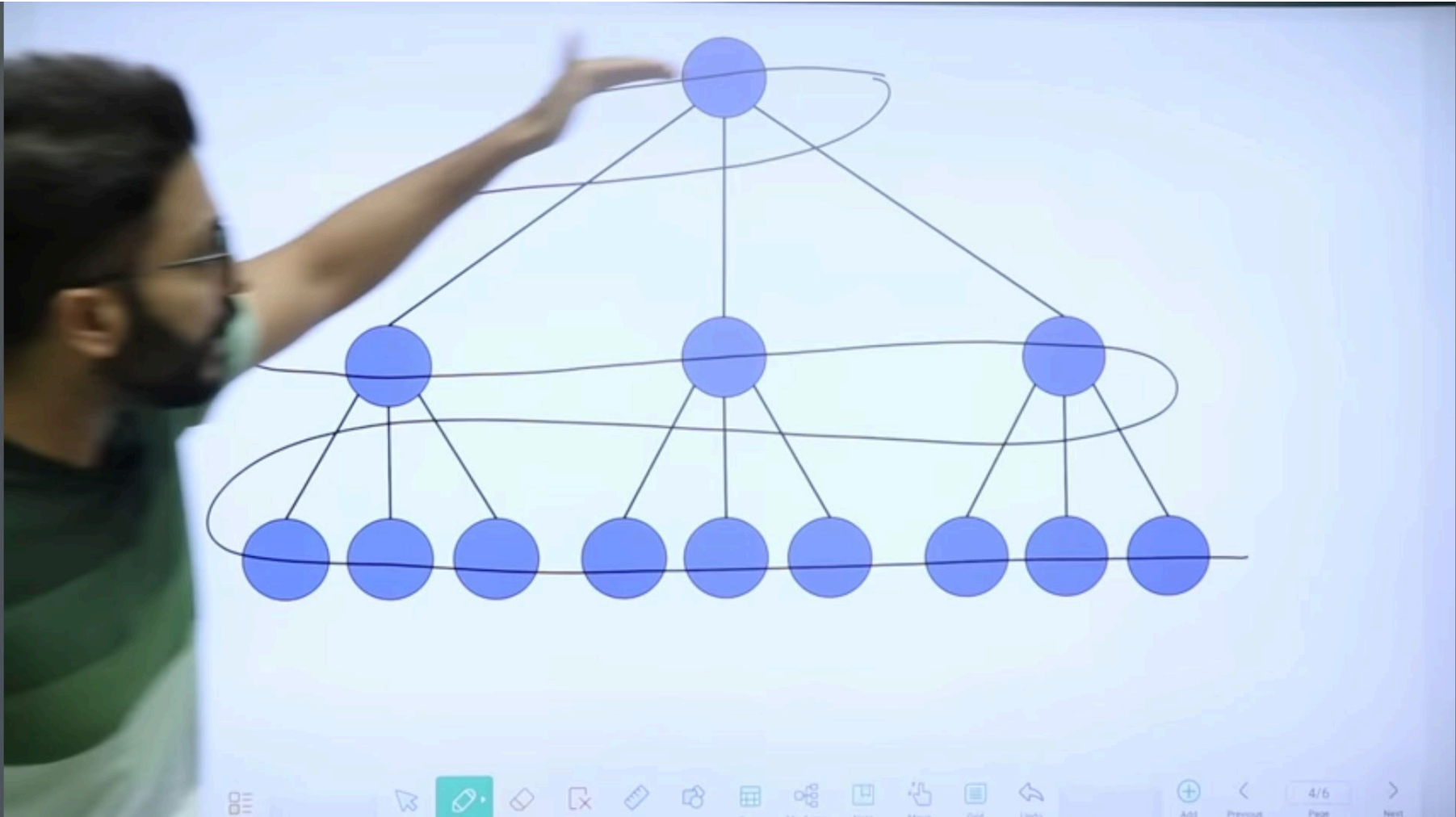


BFS - DFS

(Graph Traversal)

Introduction To Graph Traversal

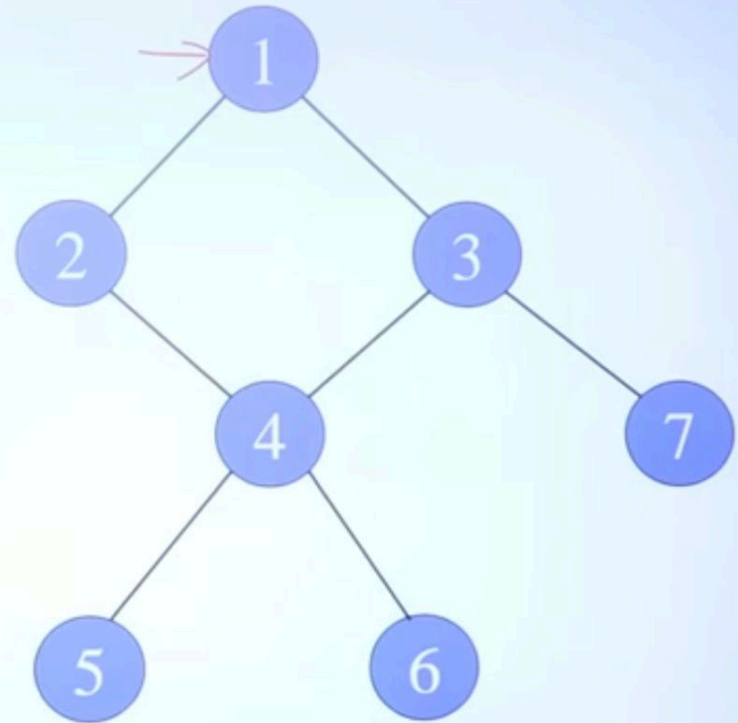
- The process of visiting and exploring a graph for processing is called graph traversal.
- Breadth First Search(BFS)
- Depth First Search(DFS)



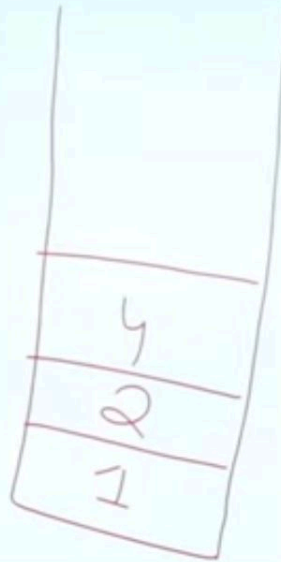
BFS

| | | | | | | |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1 | 2 | 3 | 4 | 7 | 5 | 6 |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|

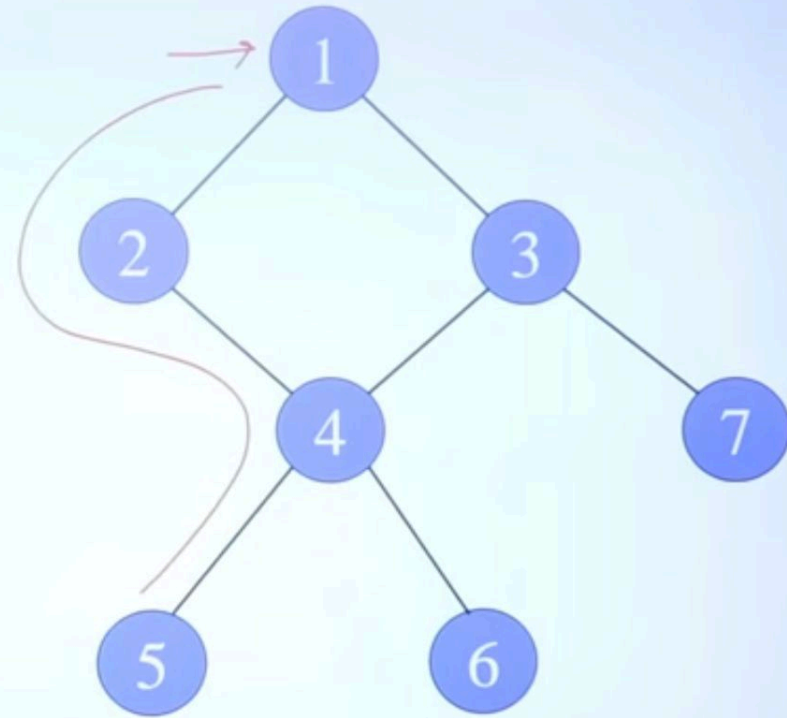
1 2 3 4 7 5 6



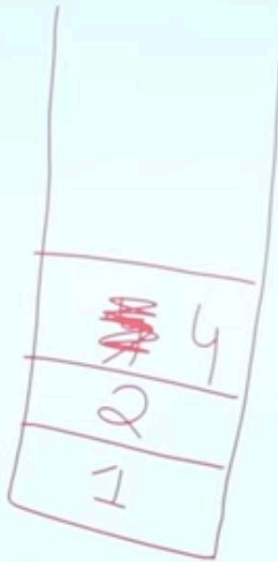
DFS



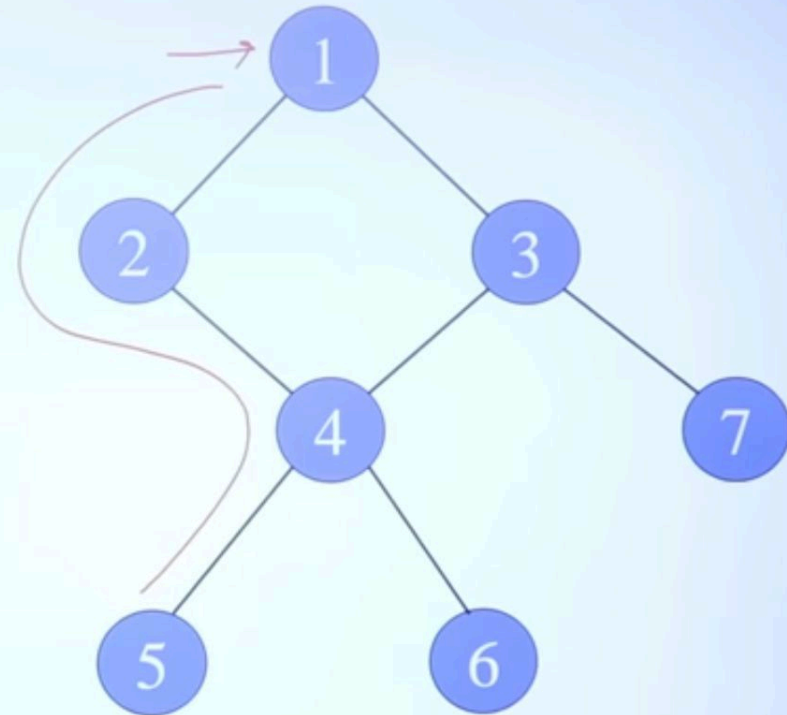
1 2 4 5



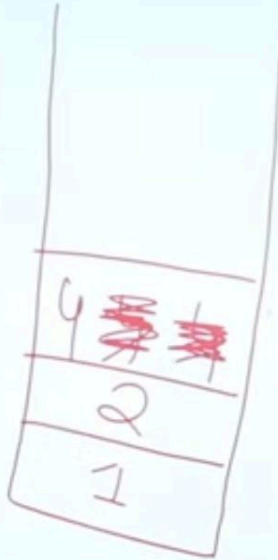
DFS



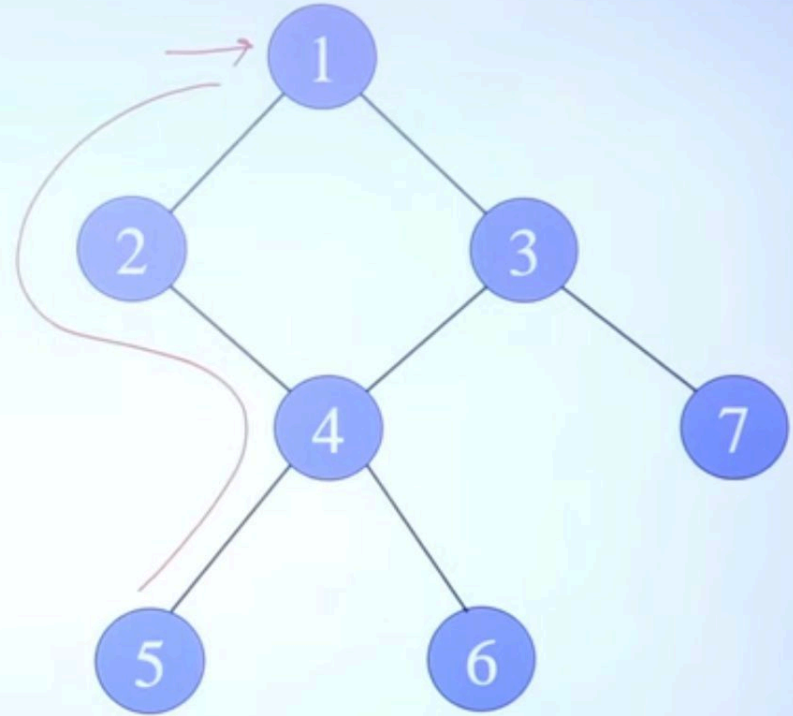
1 2 4 5 6



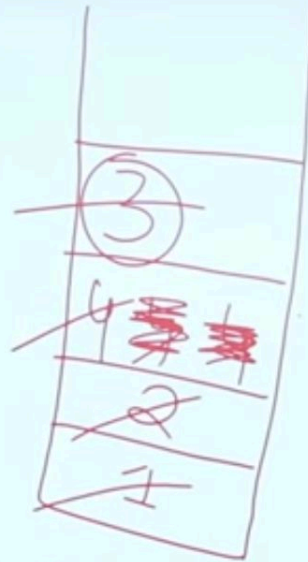
DFS



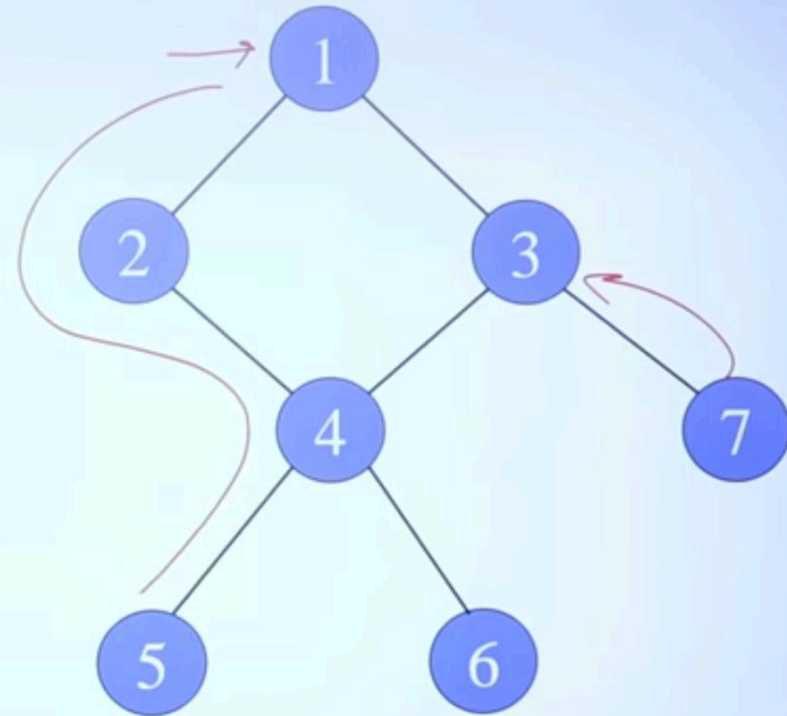
4 5 6 3



DFS



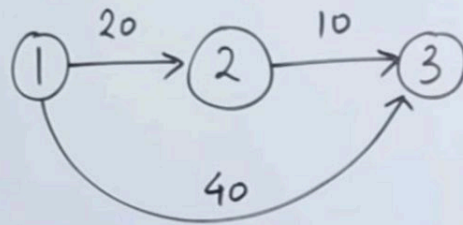
1 2 4 5 6 3 7



Dijkstras Algorithm



"Dijkstra's Algorithm" (Single Source Shortest Path)

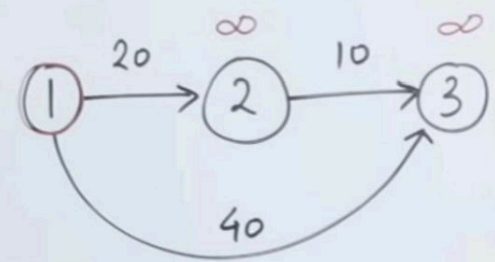


* Relaxation

$$\text{if } d(u) + c(u, v) < d(v)$$
$$d(v) = d(u) + c(u, v)$$



"Dijkstra's Algorithm" (Single Source Shortest Path)



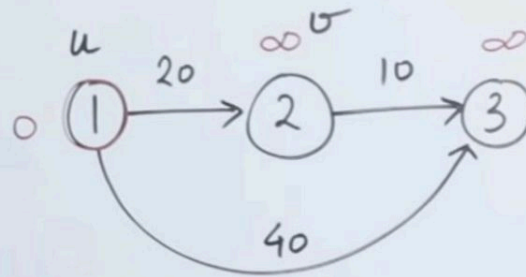
* Relaxation

$$\text{if } d(u) + c(u, v) < d(v) \\ d(v) = d(u) + c(u, v)$$





"Dijkstra's Algorithm" (Single Source Shortest Path)



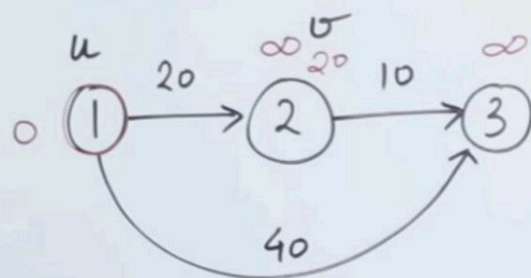
* Relaxation

$$\text{if } d(u) + c(u, v) < d(v) \\ d(v) = d(u) + c(u, v)$$

$$\begin{array}{ccccc} d(u) & + & c(u, v) & & d(v) \\ 0 & & 20 & < & \infty \end{array}$$



"Dijkstra's Algorithm" (Single Source Shortest Path)



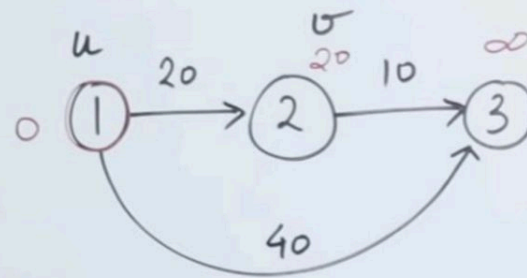
* Relaxation

$$\text{if } d(u) + c(u, v) < d(v) \\ d(v) = d(u) + c(u, v)$$

$$d(u) + c(u, v) < d(v) \\ 0 + 20 < \infty$$



"Dijkstra's Algorithm" (Single Source Shortest Path)



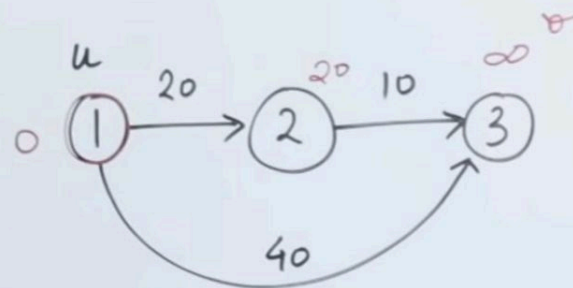
* Relaxation

$$\text{if } d(u) + c(u, v) < d(v) \\ d(v) = d(u) + c(u, v)$$

$$\begin{array}{ccc} d(u) + c(u, v) & & d(v) \\ 0 + 20 & < & \infty \end{array}$$



"Dijkstra's Algorithm" (Single Source Shortest Path)



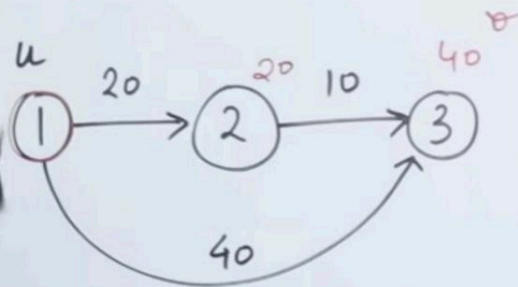
* Relaxation

$$\text{if } d(u) + C(u, v) < d(v) \\ d(v) = d(u) + C(u, v)$$

$$\begin{array}{lcl} d(u) + C(u, v) & & d(v) \\ 0 + 20 & < & \infty \\ 0 + 40 & < & \infty \\ 40 & < & \infty \end{array}$$



"Dijkstra's Algorithm" (Single Source Shortest Path)



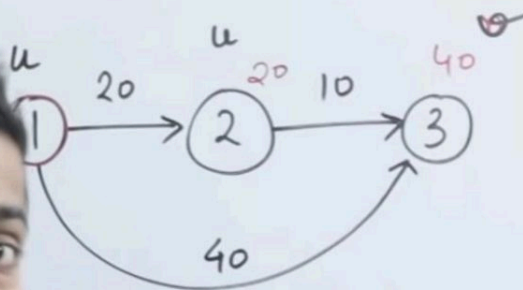
* Relaxation

$$\text{if } d(u) + c(u, v) < d(v) \\ d(v) = d(u) + c(u, v)$$

$$\begin{array}{rcl} d(u) + c(u, v) & & d(v) \\ 0 + 20 & < & \infty \\ 0 + 40 & < & \infty \\ 40 & < & \infty \end{array}$$



"Dijkstra's Algorithm" (Single Source Shortest Path)



* Relaxation

if $d(u) + c(u, v) < d(v)$
 $d(v) = d(u) + c(u, v)$

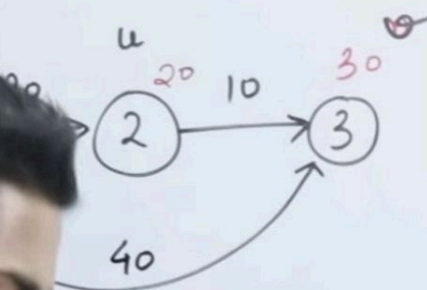
$$\begin{array}{rcl} d(u) + c(u, v) & & d(v) \\ 0 + 20 & < & \infty \\ 0 + 40 & < & \infty \\ 40 & < & \infty \end{array}$$

$$\begin{array}{l} 20 + 10 < 40 \\ 30 < 40 \end{array}$$





"Dijkstra's Algorithm" (Single Source Shortest Path)

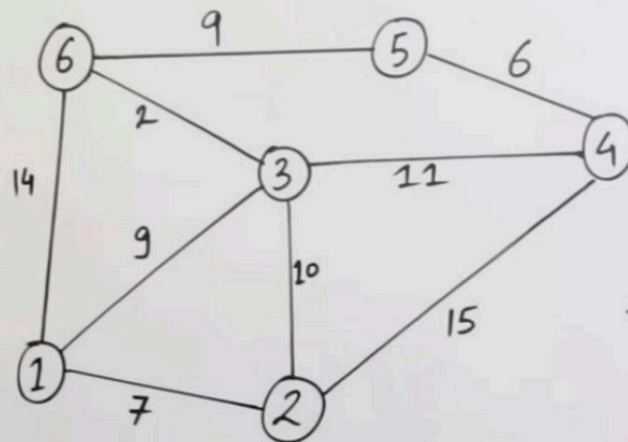


* Relaxation

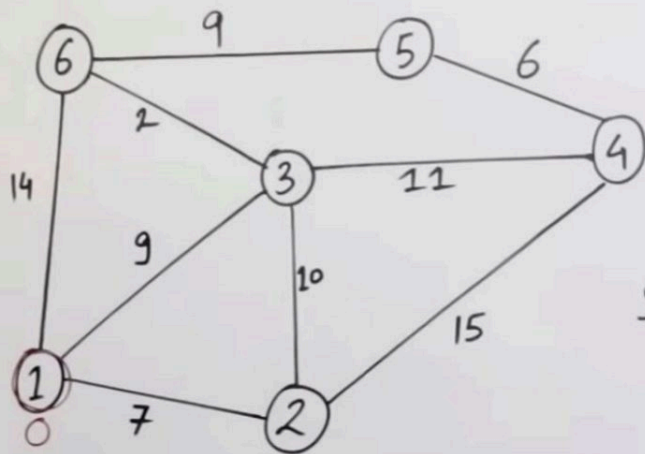
$$\text{if } d(u) + c(u, v) < d(v) \\ d(v) = d(u) + c(u, v)$$

$$\begin{array}{lcl} d(u) + c(u, v) & & d(v) \\ 0 + 20 & < & \infty \\ 0 + 40 & < & \infty \\ 40 & < & \infty \end{array}$$

$$\begin{array}{l} 20 + 10 < 40 \\ 30 < 40 \end{array}$$

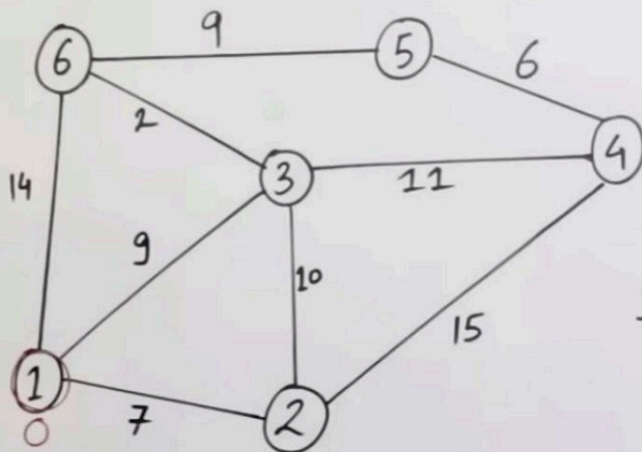


| Source | Destination | | | | |
|--------|-------------|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| | | | | | |



| Source 1 | Destination | | | | |
|-------------|-------------|----------|----------|----------|----------|
| | 2 | 3 | 4 | 5 | 6 |
| | ∞ | ∞ | ∞ | ∞ | ∞ |
| | 7 | 9 | ∞ | ∞ | 14 |





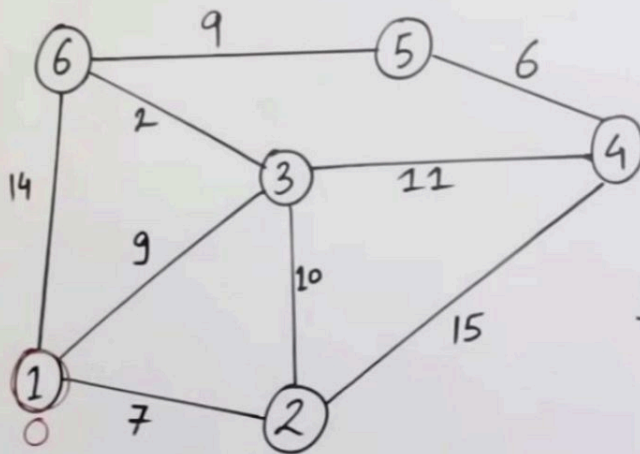
| Source | Destination | | | | |
|--------|-------------|----------|----------|----------|----------|
| 1 | 2 | 3 | 4 | 5 | 6 |
| | ∞ | ∞ | ∞ | ∞ | ∞ |
| | 7 | 9 | ∞ | ∞ | 14 |
| 1,2 | 7 | 9 | 22 | ∞ | 14 |



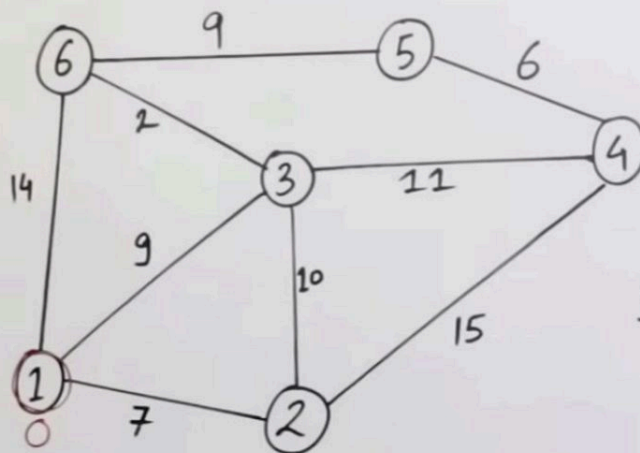
13/52

SUBSCRIBED





| Source | Destination | | | | |
|--------|-------------|----------|----------|----------|----------|
| 1 | 2 | 3 | 4 | 5 | 6 |
| | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1,2 | 7 | 9 | ∞ | ∞ | 14 |
| 1,2,3 | 7 | 9 | 22 | ∞ | 14 |
| | 7 | 9 | 20 | ∞ | 11 |

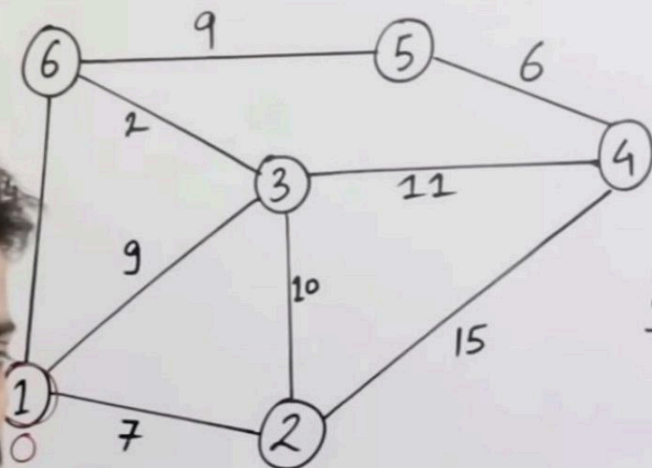


1-3-6-5

1,2
1,2,3
1,2,3,6

| Source 1 | Destination | | | | |
|-------------|-------------|----------|----------|----------|----------|
| | 2 | 3 | 4 | 5 | 6 |
| | ∞ | ∞ | ∞ | ∞ | ∞ |
| | 7 | 9 | ∞ | ∞ | 14 |
| | 7 | 9 | 22 | ∞ | 14 |
| | 7 | 9 | 20 | ∞ | 11 |
| | 7 | 9 | 20 | 20 | 11 |





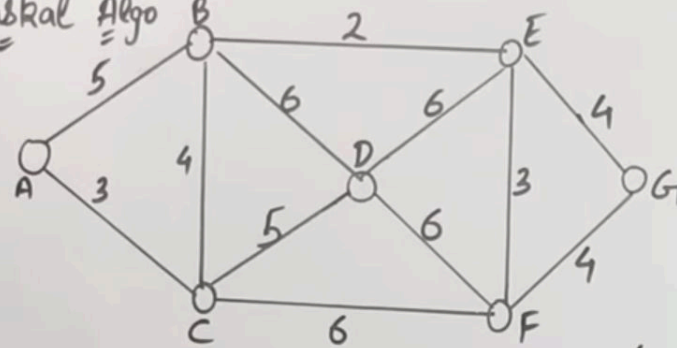
1-3-6-5

- 1,2
- 1,2,3
- 1,2,3,6
- 1,2,3,6,4
- 1,2,3,6,4,5

| Source 1 | Destination | | | | |
|-------------|-------------|----------|----------|----------|----------|
| | 2 | 3 | 4 | 5 | 6 |
| | ∞ | ∞ | ∞ | ∞ | ∞ |
| | 7 | 9 | ∞ | ∞ | 14 |
| | 7 | 9 | 22 | ∞ | 14 |
| | 7 | 9 | 20 | ∞ | 11 |
| | 7 | 9 | 20 | 20 | 11 |
| | | | | 20 | |

Kruskal's Algorithm

'Kruskal Algo'



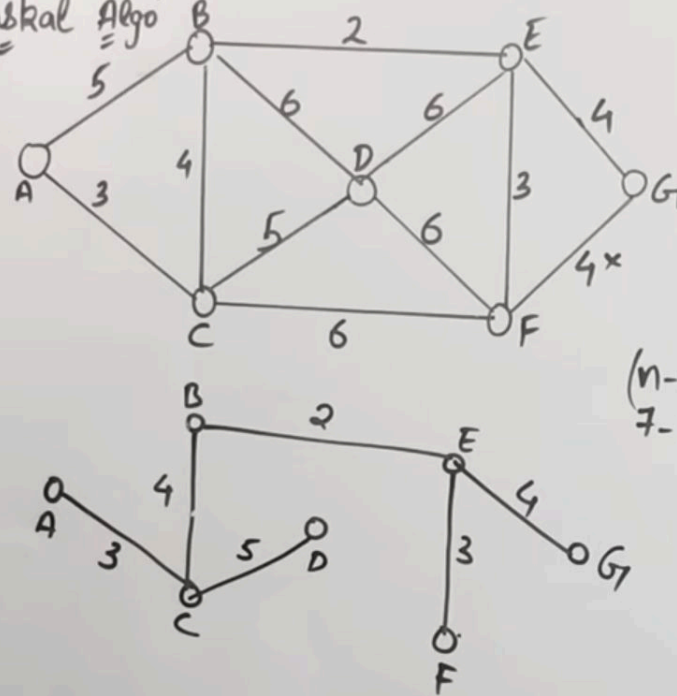
- 1) Construct Min Heap with 'e' edges
- 2) Take one by one edge and add in spanning tree. (Cycle should not be created)

Best Case $(n-1)$ edges

Worst Case 'e' edges

$$(n-1) \\ 7-1 = \underline{6}$$

'Kruskal Algo'



$$(n-1) \\ 7-1=6$$

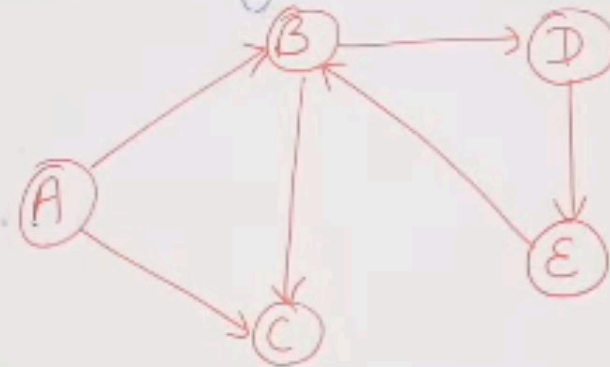
- 1) Construct Min Heap with 'e' edges
- 2) Take one by one edge and add in spanning tree. (Cycle should not be created)

Best Case $(n-1)$ edges

Worst Case 'e' edges

Cycle detection (Directed graph)

Detect Cycle in Directed Graph using DFS



flag:

- 1 = Unvisited
- 0 = Visited & in Stack
- 1 = Visited & Popped out from stack

Stack:-

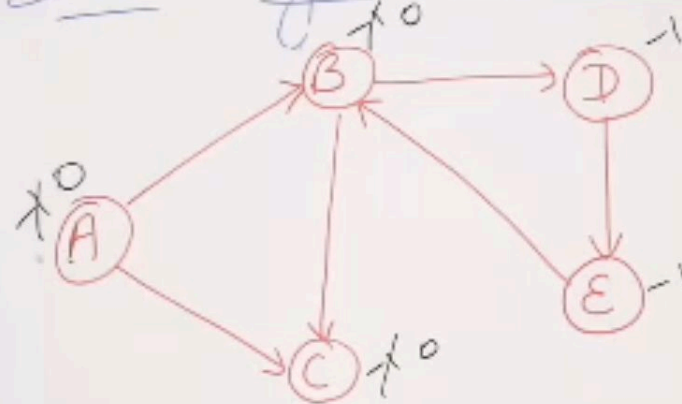
Visited Set:-

Parent Map

| Vertex | Parent |
|--------|--------|
| | |
| | |
| | |

Subscribe JENNY'S LECTURES CS/IT NET&JRF for full syllabus

Detect Cycle in Directed Graph using DFS



flag:

- 1 = Unvisited
- 0 = Visited & in Stack
- 1 = Visited & Popped out from stack

Stack:-

| |
|---|
| |
| |
| C |
| B |
| A |

Visited Set:-

A B C

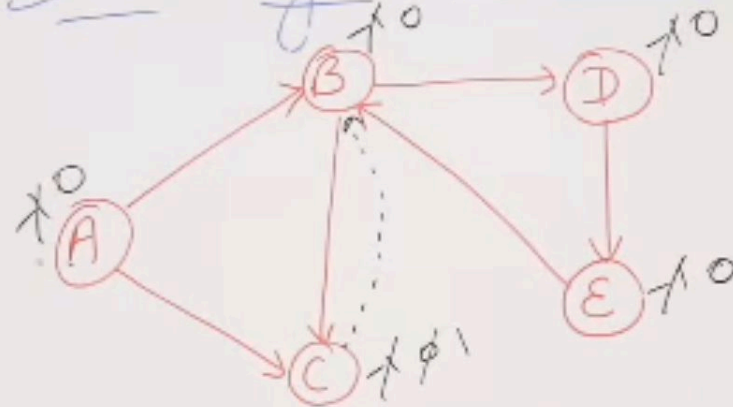
Parent Map

| Vertex | Parent |
|--------|--------|
| A | - |
| B | A |
| C | B |

Subscribe JENNY'S LECTURES CS/IT NET&JRF for full syllabus



Detect Cycle in Directed Graph using DFS



flag:

- 1 = Unvisited
- 0 = Visited & in Stack
- 1 = Visited & Popped out from stack

Stack:-

| |
|---|
| |
| |
| E |
| D |
| B |
| A |

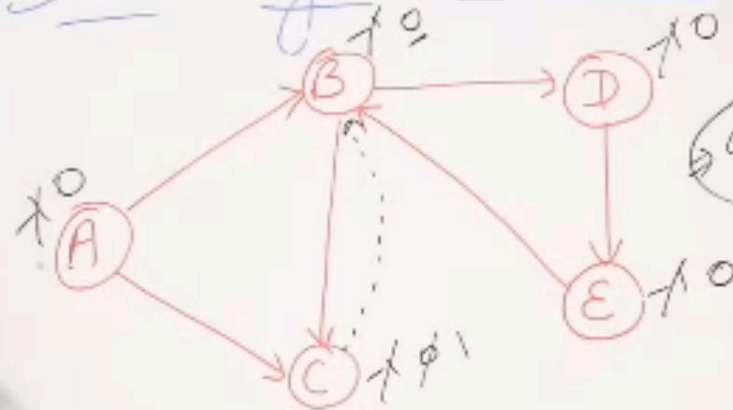
Visited Set:-

A B C D E

Parent Map

| Vertex | Parent |
|--------|--------|
| A | - |
| B | A |
| C | B |
| D | B |
| E | D |

Detect Cycle in Directed Graph using DFS



flag:
 -1 = Unvisited
 0 = Visited & in Stack
 1 = Visited & Popped out from stack

ack:-

| |
|---|
| |
| E |
| D |
| B |
| A |

Visited Set:-

A B C D E

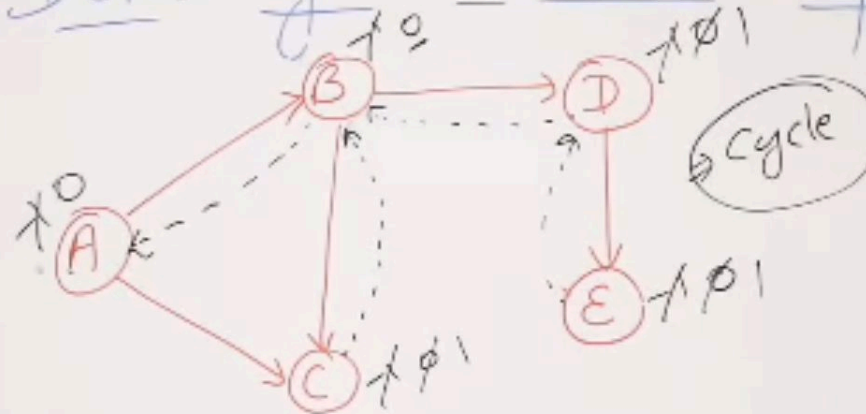
B → D → E → B

Parent Map

| Vertex | Parent |
|--------|--------|
| A | - |
| B | A |
| C | B |
| D | B |
| E | D |

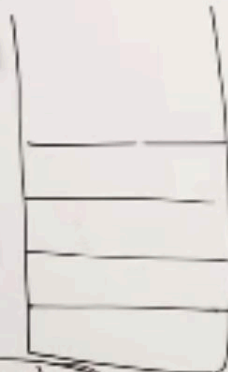


Detect Cycle in Directed Graph using DFS



flag:
 - 1 = Unvisited
 0 = Visited & in Stack
 1 = Visited & Popped out from stack

Stack:-



empty

Visited Set:-

A B C D E

B → D → E → B

Parent Map

| Vertex | Parent |
|--------|--------|
| A | - |
| B | A |
| C | B |
| D | B |
| E | D |

