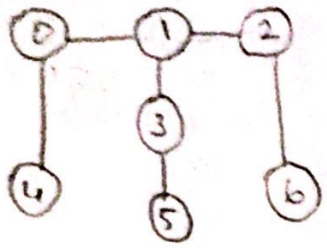
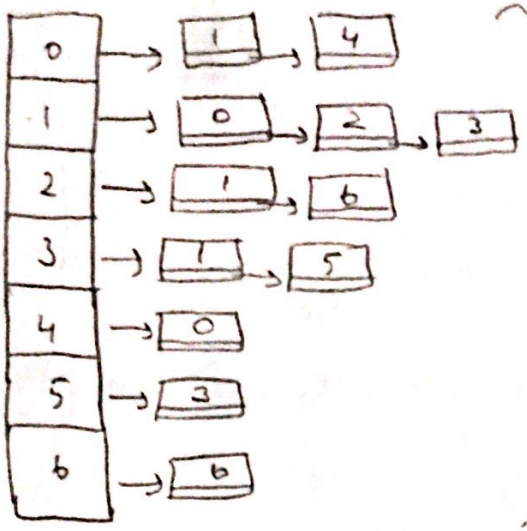


Represent adjacency list



Array of list of this structure

Nodes are considered bilateral because it is a non directional graph.

Represent adjacency matrix

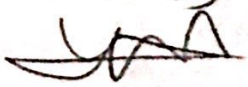
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]		1			1		
[1]	1		1	1			
[2]		1					1
[3]		1				1	
[4]	1						
[5]				1			
[6]			1				

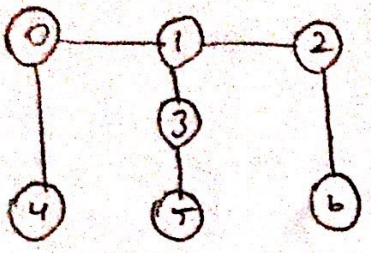
Source is represented rows and destination is represented columns.

Row

Column

Muharrem Ozan Yeşiller
171044033



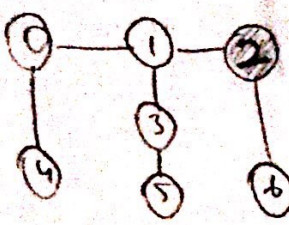


DFS (start 2)

(0 unvisited, O visited, being visited)

(DO = discovery order, FO = finish order)

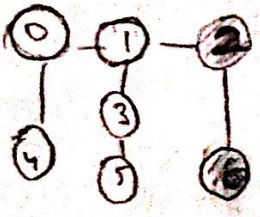
1)



DO = 2

FO = ∅

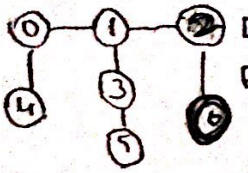
2)



DO = 2, 6

FO = ∅

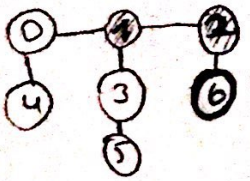
3)



DO = 2, 6

FO = 6

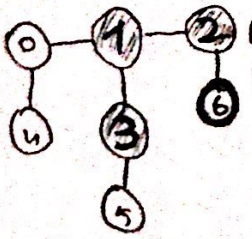
4)



DO = 2, 6, 1

FO = 6

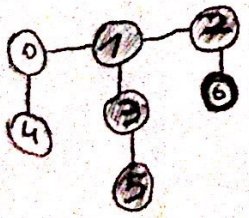
5)



DO = 2, 6, 1, 3

FO = 6

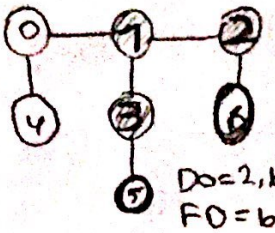
6)



DO = 2, 6, 1, 3, 5

FO = 6

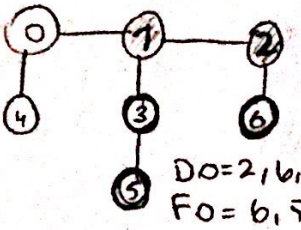
7)



DO = 2, 6, 1, 3, 5

FO = 6, 5

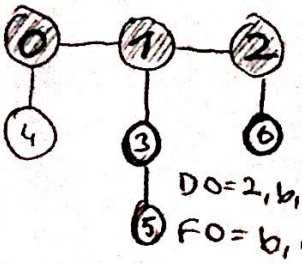
8)



DO = 2, 6, 1, 3, 5

FO = 6, 5, 3

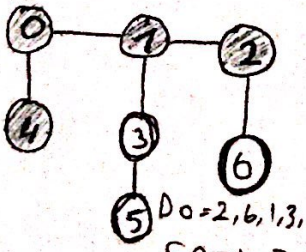
9)



DO = 2, 6, 1, 3, 5, 0

FO = 6, 5, 3

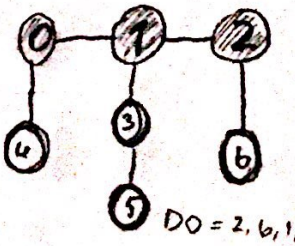
10)



DO = 2, 6, 1, 3, 5, 0, 4

FO = 6, 5, 3

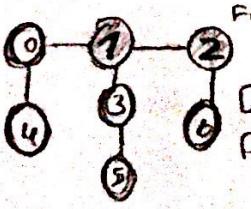
11)



DO = 2, 6, 1, 3, 5, 0, 4

FO = 6, 5, 3, 4

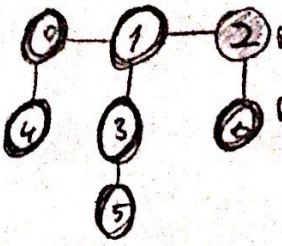
12)



DO = 2, 6, 1, 3, 5, 0, 4

FO = 6, 5, 3, 4, 0

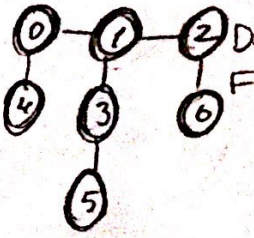
13)



DO = 2, 6, 1, 3, 5, 0, 4

FO = 6, 5, 3, 4, 0, 1

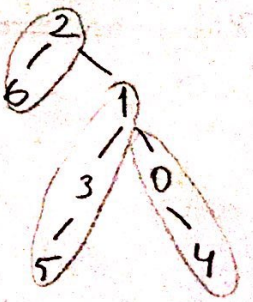
14)



DO = 2, 6, 1, 3, 5, 0, 4

FO = 6, 5, 3, 4, 0, 1, 2

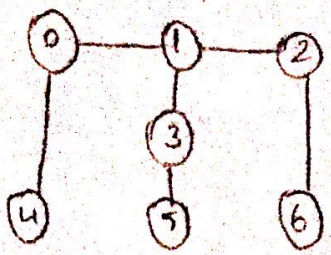
we can represent like tree result



Discovery (visit order) = 2, 6, 1, 3, 5, 0, 4

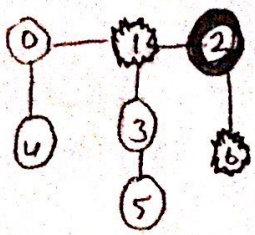
Finish order = 6, 5, 3, 4, 0, 1, 2

Muharrem Ozan Yesiller
171044033

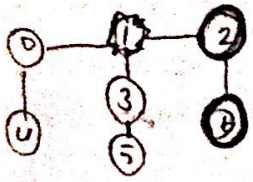


BFS (start = 2)

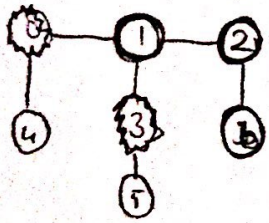
visited
un visited
identified

① 

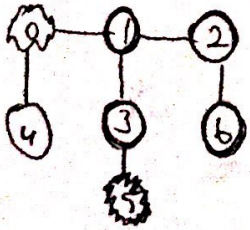
queue = 1, 6
visited = 2

② 

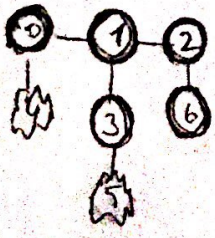
queue = 1
visited = 6, 2

③ 

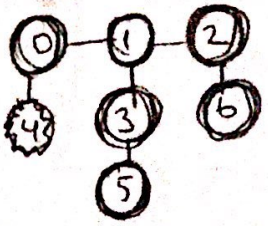
queue = 0, 3
visited = 1, 6, 2

④ 

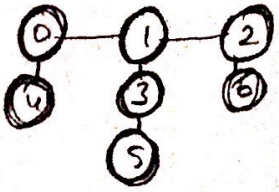
queue = 5, 0
visited = 3, 1, 6, 2

⑤ 

queue = 4, 5
visited = 0, 3, 1, 6, 2

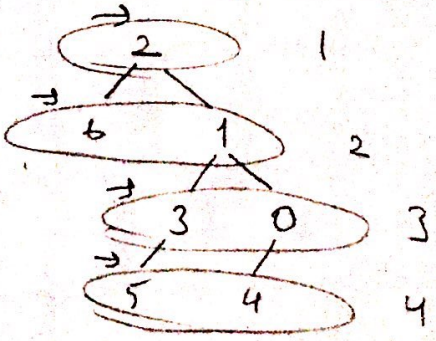
⑥ 

queue = 4
visited = 5, 0, 3, 1, 6, 2

⑦ 

queue = empty
visited = all of vertex

we can represent like tree, this graph (start 2)

=> 

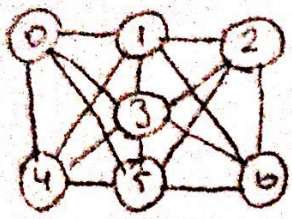
2, 6, 1, 3, 0, 5, 4

visit order

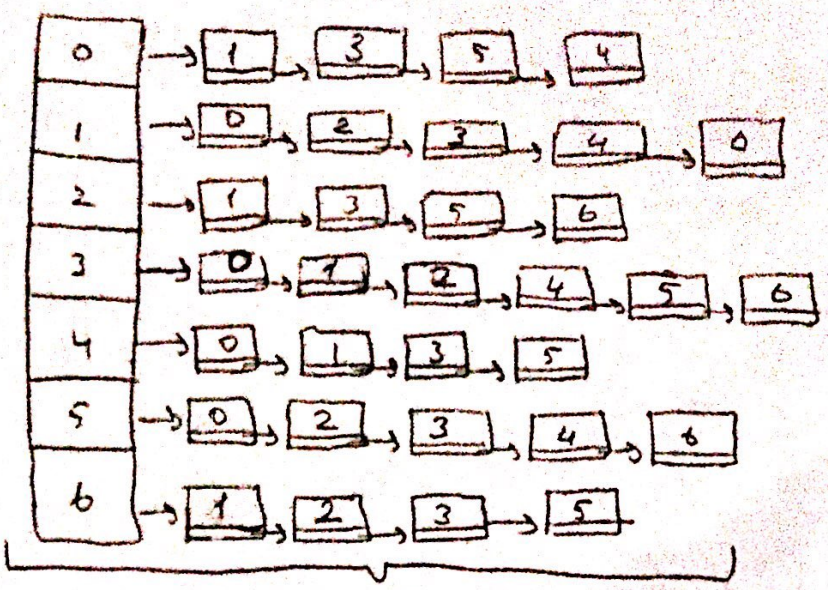
the result is array that index is destination and value of index is source

[0] = 1
[1] = 2
[2] = -1
[3] = 1
[4] = 0
[5] = 3
[6] = 2

Muharrem Ozan Yesiller
171044033



Represent adjacency list



Array of list of this structure
Nodes are considered bilateral because it is a non directional graph

Represent adjacency matrix

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]		1		1	1	1	
[1]	1		1	1	1		1
[2]		1		1		1	1
[3]	1	1	1		1	1	1
[4]	1	1		1		1	
[5]	1		1	1	1		1
[6]		1	1	1		1	

Source is represented rows and destination is represented columns

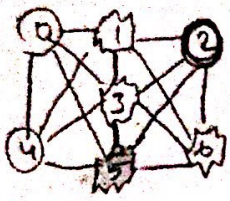
Row

column

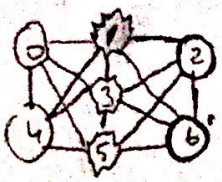
Muharrem Ozan Yesiller
171044033

BFS (start=2)

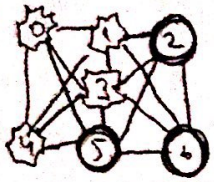
visited
unvisited
identified

① 

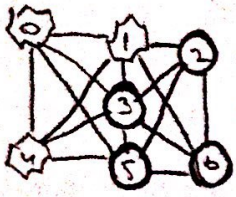
queue = 1, 3, 5, 6
visited = 2

② 

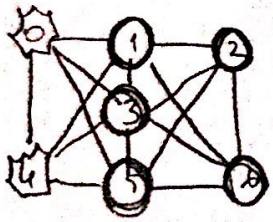
queue = 1, 3, 5
visited = 6, 2

③ 

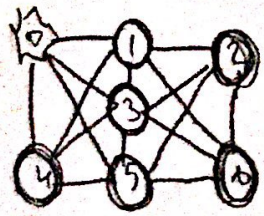
queue = 0, 4, 1, 3
visited = 5, 6, 2

④ 

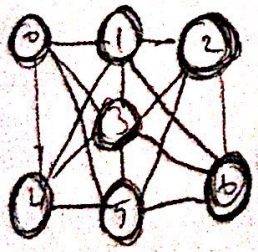
queue = 0, 4, 1
visited = 3, 5, 6, 2

⑤ 

queue = 0, 4
visited = 1, 3, 5, 6, 2

⑥ 

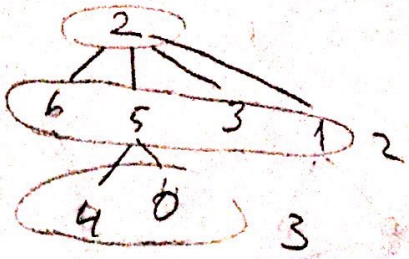
queue = 0
visited = 4, 1, 3, 5, 6, 2

⑦ 

queue = empty
visited = all of vertex

we can represent like tree this graph (start 2)

=>

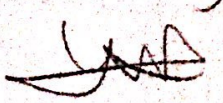


2, 6, 5, 3, 1, 4, 0,
visit order

the result is array that index is destination and value of index is source

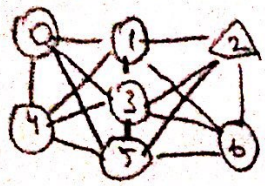
[0] = 5
[1] = 2
[2] = -1
[3] = 2
[4] = 5
[5] = 2
[6] = 2

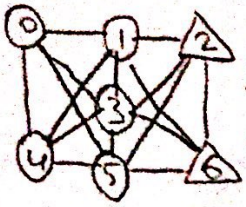
Muharrem Ozan Yesiller
171044033

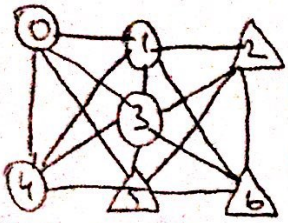


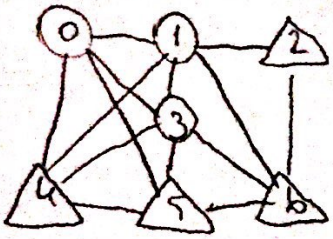
DFS (start 2)

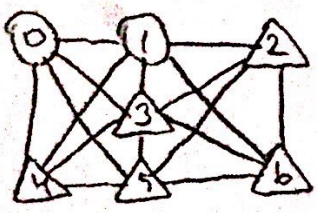
(○ unvisited, ◯ visited, △ being visited) (DO = discovery order, FO = finish order)

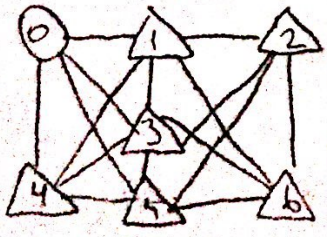
1)

DO = 2
FO = ∅

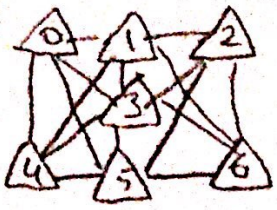
2)

DO = 2, 6
FO = ∅

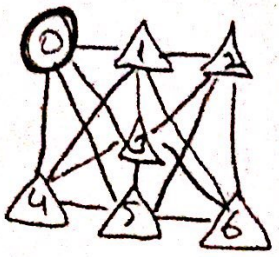
3)

DO = 2, 6, 5
FO = ∅

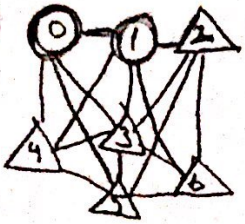
4)

DO = 2, 6, 5, 4
FO = ∅

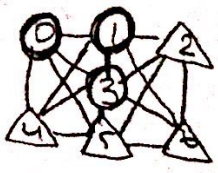
5)

DO = 2, 6, 5, 4, 3
FO = ∅

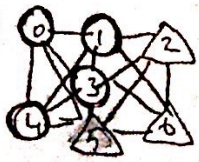
6)

DO = 2, 6, 5, 4, 3, 1
FO = ∅

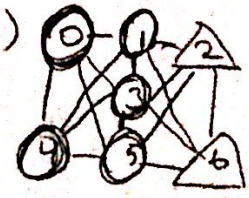
7)

DO = 2, 6, 5, 4, 3, 1, 0
FO = ∅

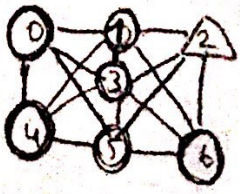
8)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0

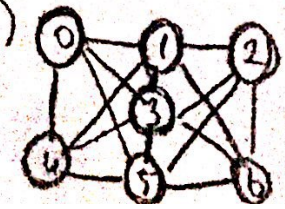
9)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0, 1

10)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0, 1, 3

11)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0, 1, 3, 4

12)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0, 1, 3, 4, 5

13)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0, 1, 3, 4, 5, 6

14)

DO = 2, 6, 5, 4, 3, 1, 0
FO = 0, 1, 3, 4, 5, 6, 2

Muharrem Ozan Yeşiller
171044033