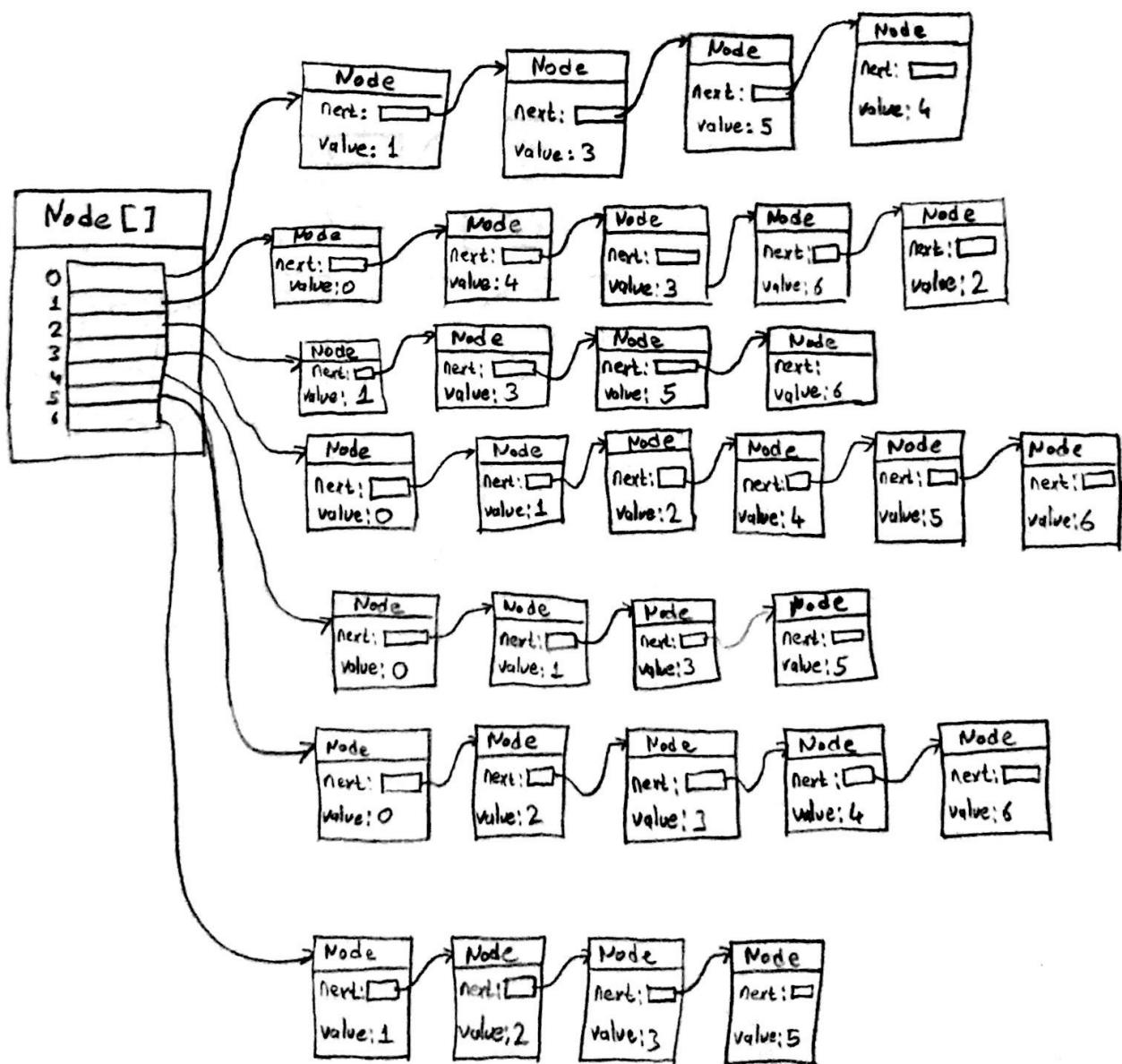
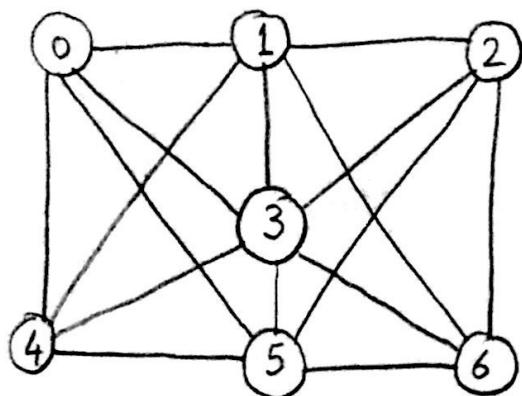
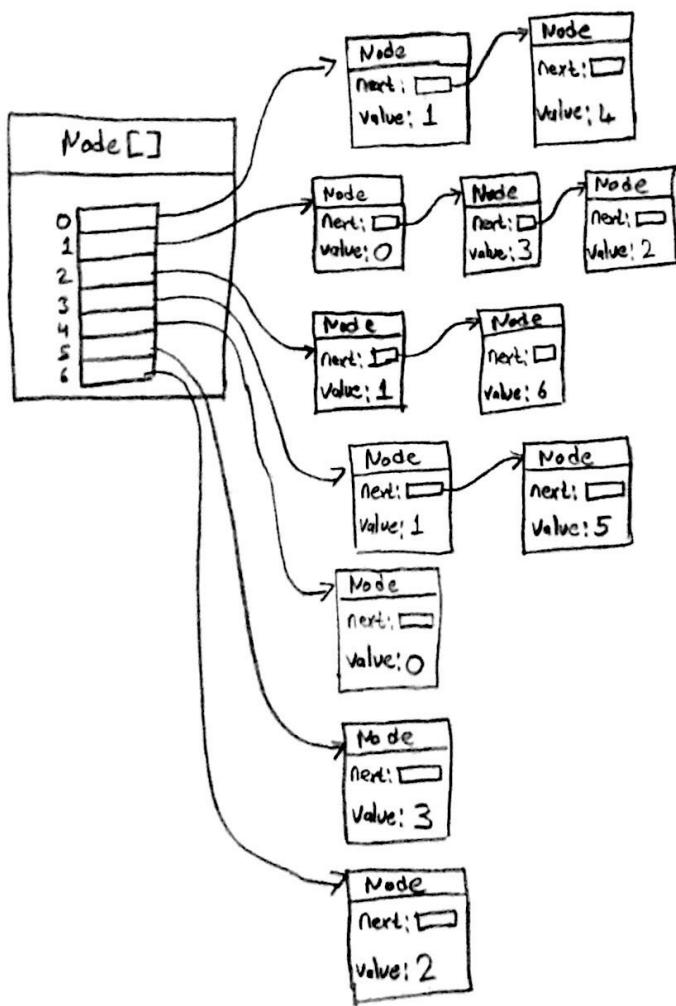
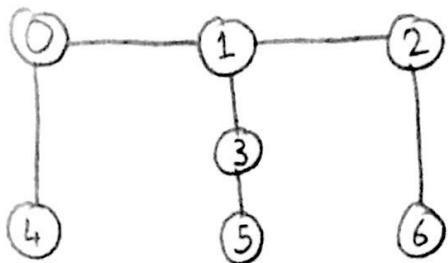


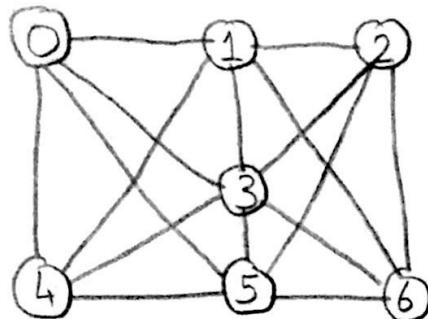
1st Graph — Adjacency List



2nd Graph - Adjacency List

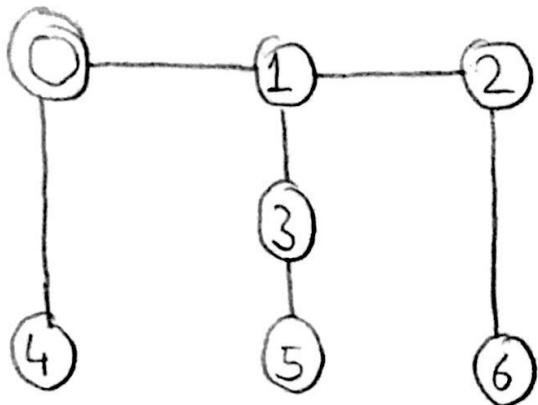


1st Graph - Adjacency Matrix



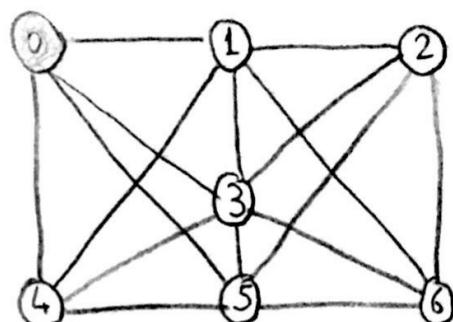
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]	1.0			1.0	1.0	1.0	
[1]	1.0	1.0		1.0	1.0		1.0
[2]	1.0	1.0	1.0	1.0		1.0	1.0
[3]	1.0	1.0	1.0	1.0	1.0	1.0	1.0
[4]	1.0	1.0	1.0	1.0	1.0		1.0
[5]	1.0	1.0	1.0	1.0	1.0	1.0	
[6]	1.0	1.0	1.0	1.0	1.0	1.0	

2nd Graph - Adjacency Matrix



	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]	1.0				1.0		
[1]	1.0	1.0	1.0	1.0			
[2]		1.0	1.0				1.0
[3]		1.0	1.0	1.0			1.0
[4]	1.0		1.0	1.0	1.0		
[5]		1.0	1.0	1.0	1.0	1.0	
[6]			1.0	1.0	1.0	1.0	1.0

Breadth -First Search of 1st Graph

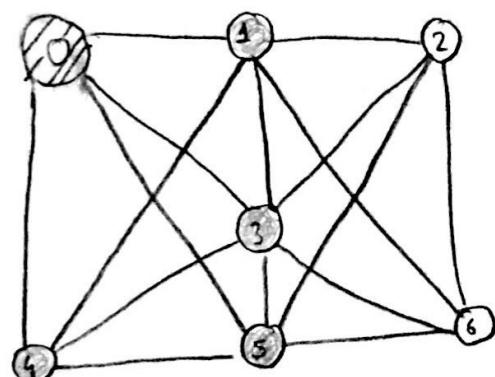


○ → in Queue

○ → Visited

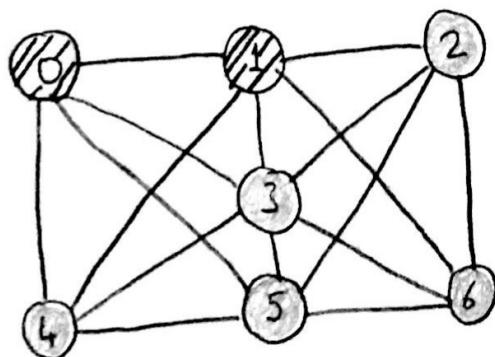
Queue: 0

Visited:



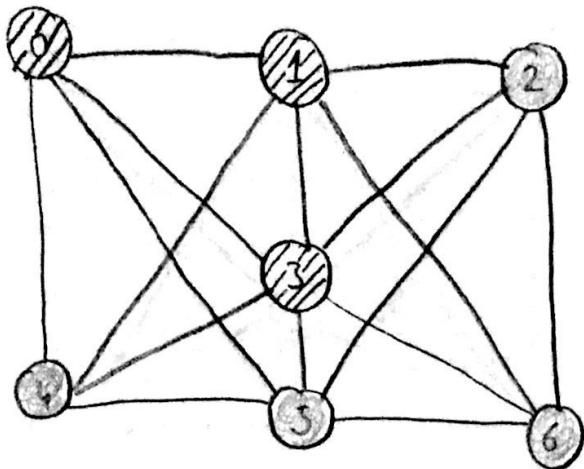
Queue: 1 3 4 5

Visited: 0



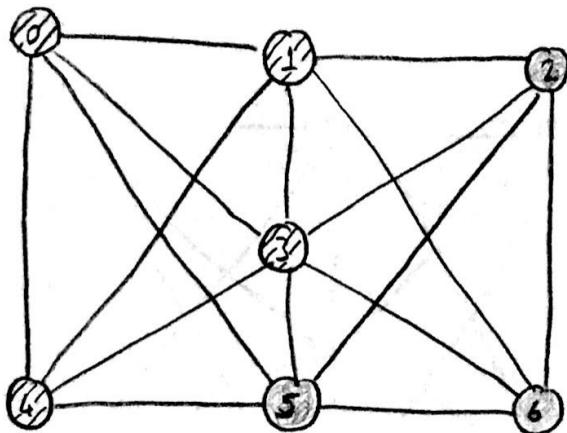
Queue: 2 6

Visited: 0 1



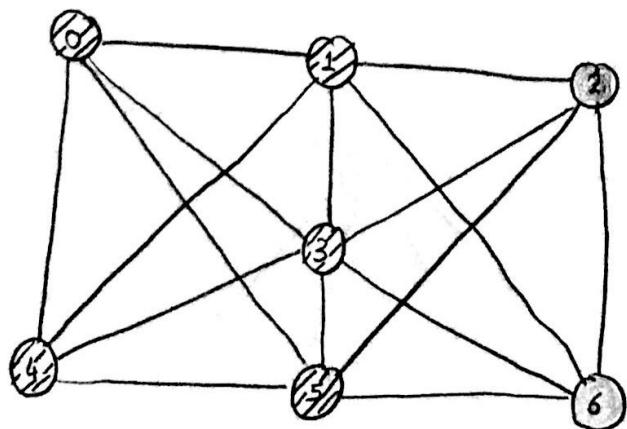
Queue: 4 5 2 6

Visited: 0 1 3



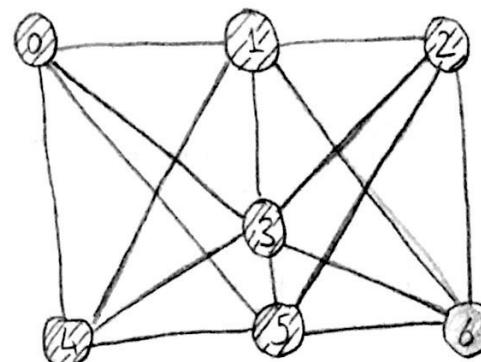
Queue: 5 2 6

Visited: 0 1 3 4



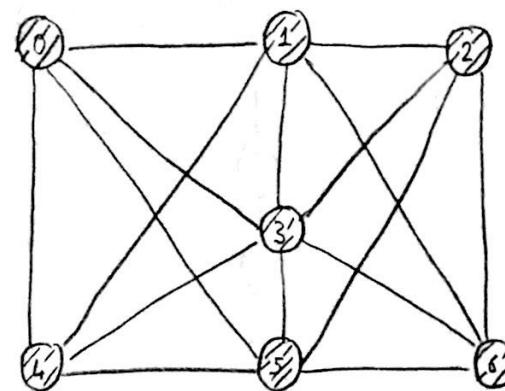
Queue: 2 6

Visited: 0 1 3 4 5



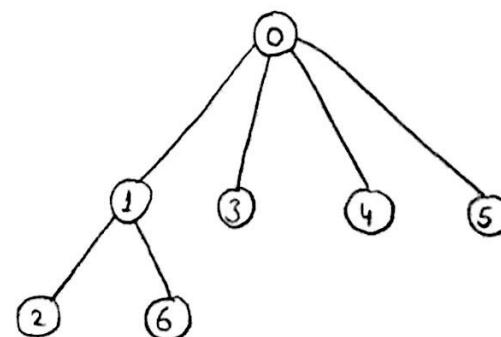
Queue: 6

Visited: 0 1 3 4 5 2

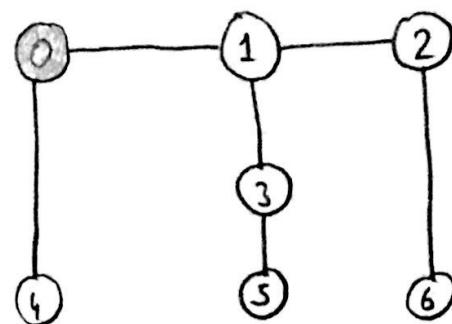


Queue: Empty

Visited: 0 1 3 4 5 2 6



Breadth-First Search of 2nd Graph

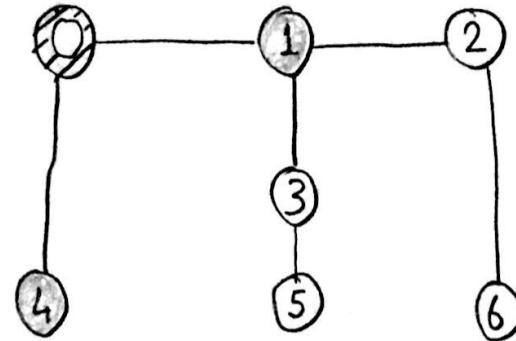


→ in Queue

→ Visited

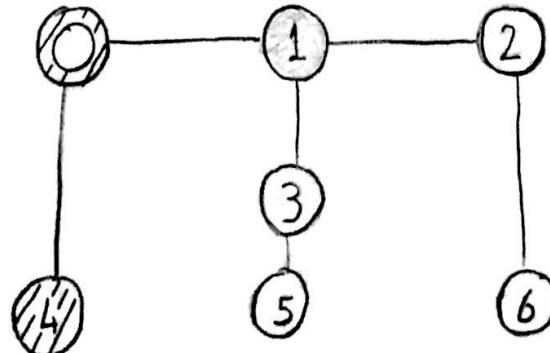
Queue: 0

Visited:



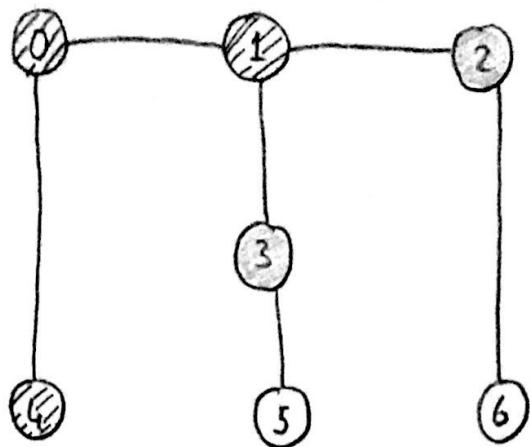
Queue: 4 1

Visited: 0



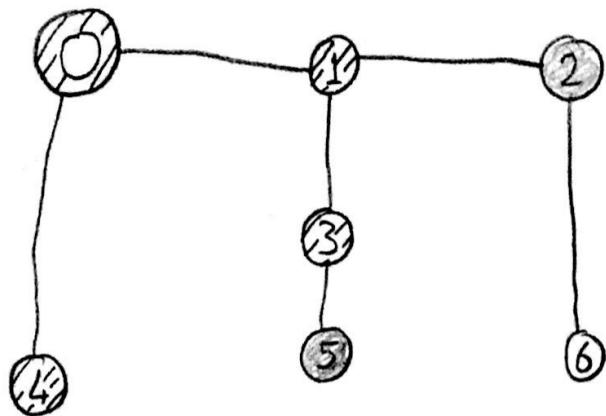
Queue: 1

Visited: 0 4



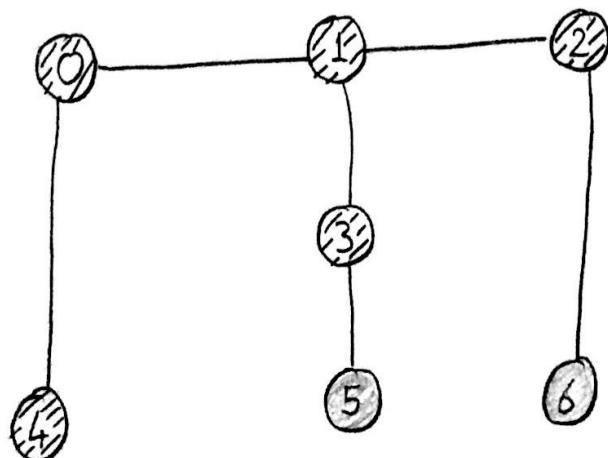
Queue: 3 2

Visited: 0 4 1



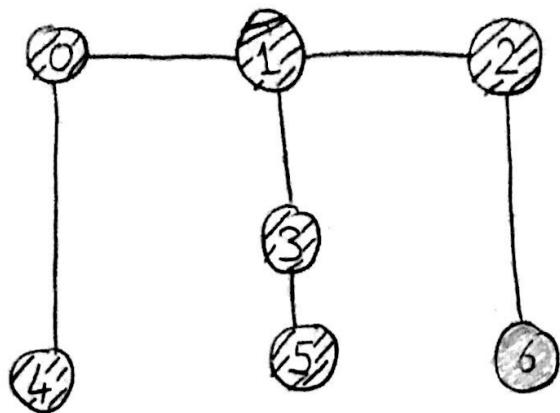
Queue: 2 5

Visited: 0 4 1 3



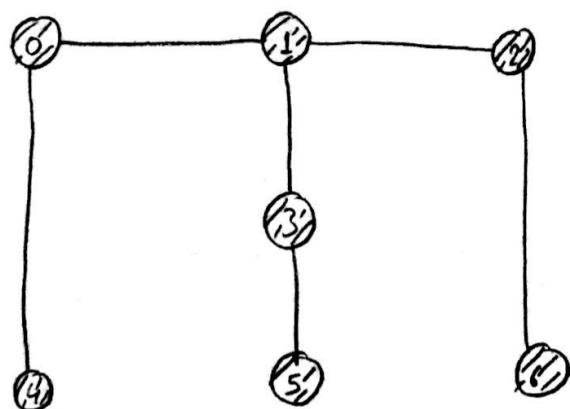
Queue: 5 6

Visited: 0 4 1 3 2



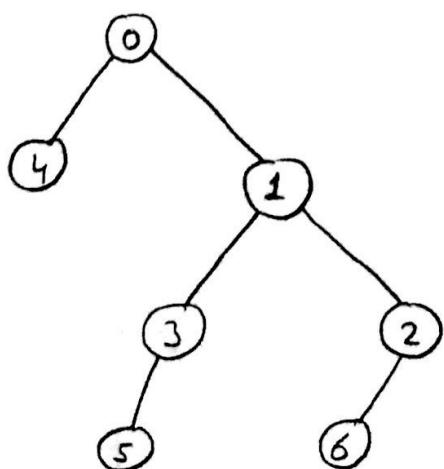
Queue: 6

Visited: 0 4 1 3 2

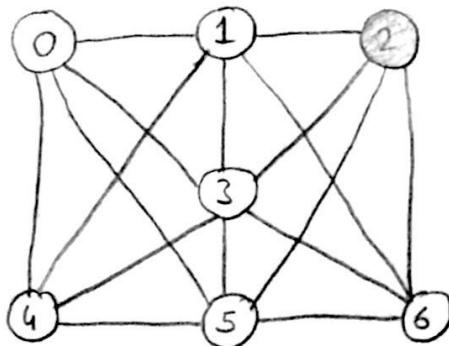


Queue: empty

Visited: 0 4 1 3 2 6



1st Graph Deep-First Search

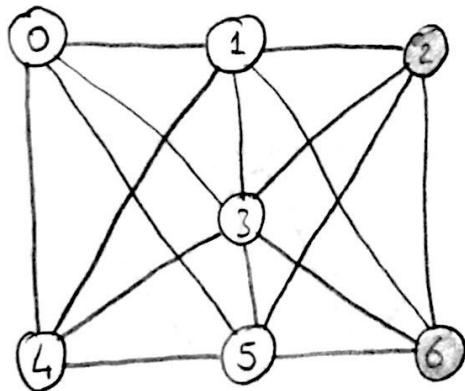


$\textcircled{X} \rightarrow \text{DVO}$

$\textcircled{\cancel{X}} \rightarrow \text{FO}$

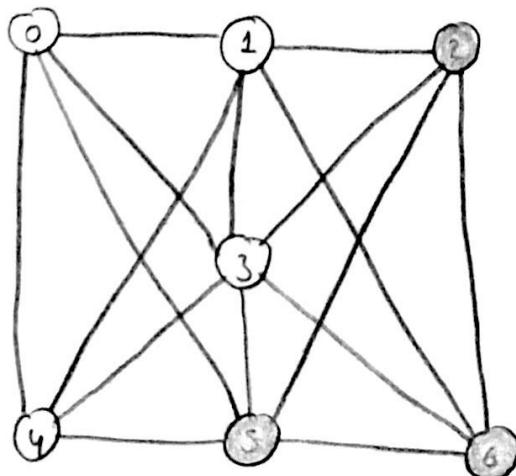
$\text{DVO} = \{2\}$

$\text{FO} = \{3\}$



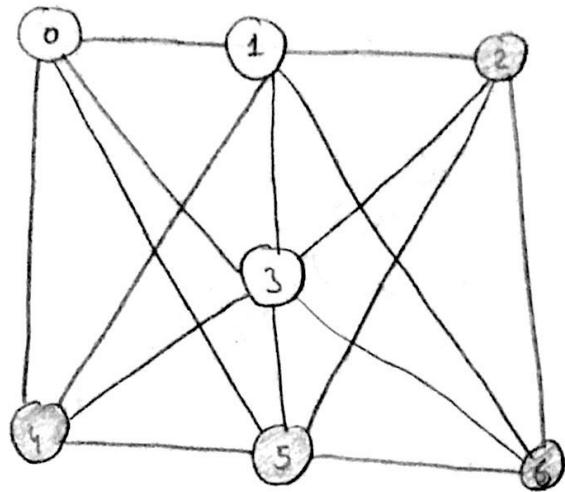
$\text{DVO} = \{2, 6\}$

$\text{FO} = \{3\}$

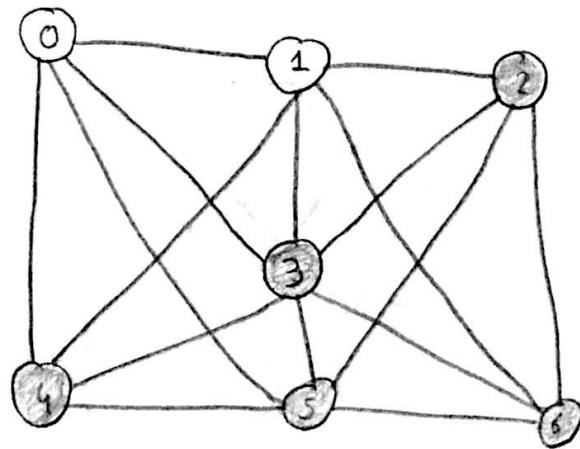


$\text{DVO} = \{2, 6, 5\}$

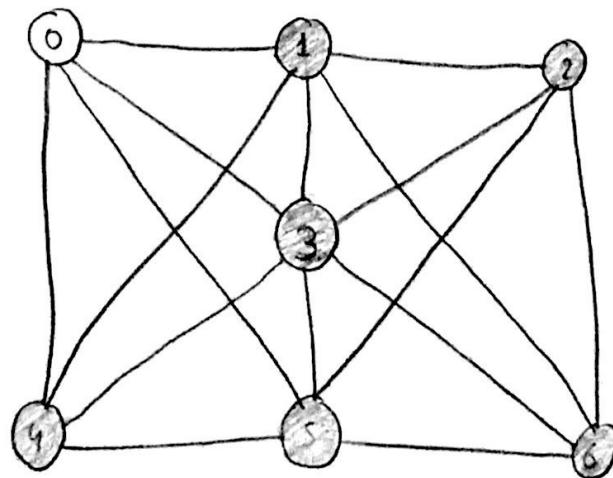
$\text{FO} = \{3\}$



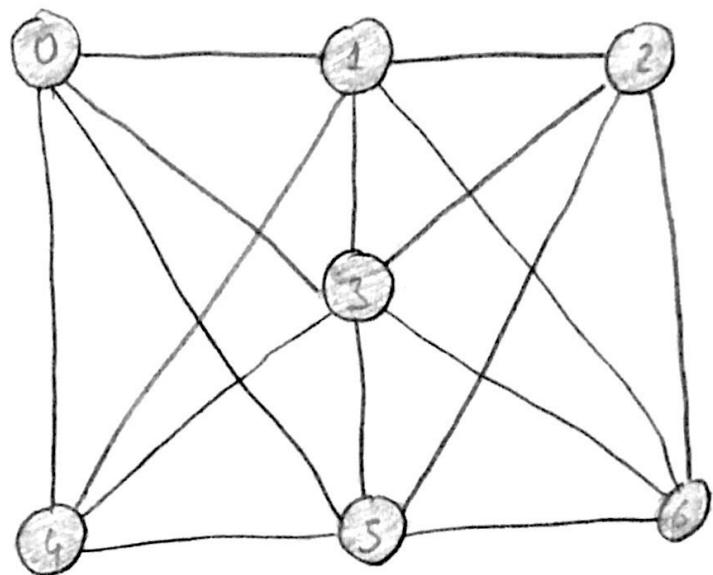
DVO = { 2, 6, 5, 4 }



DVO = { 2, 6, 5, 4, 3 }
FO = { }

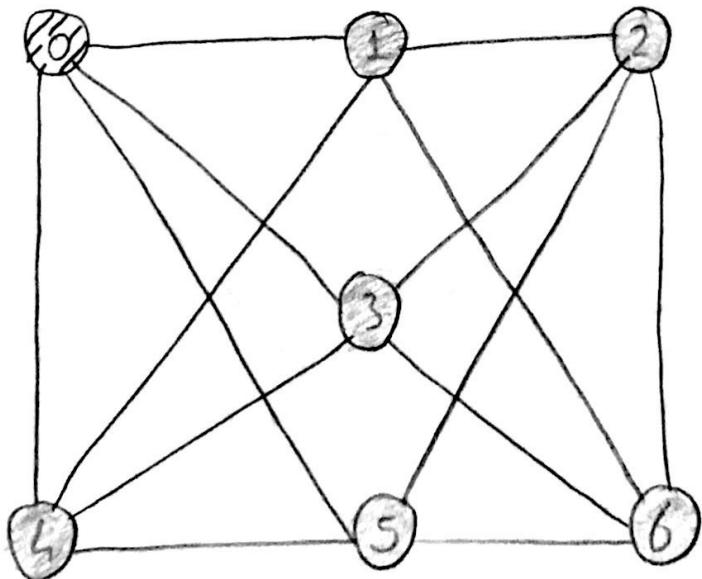


DVO = { 2, 6, 5, 4, 3, 1 }
FO = { }



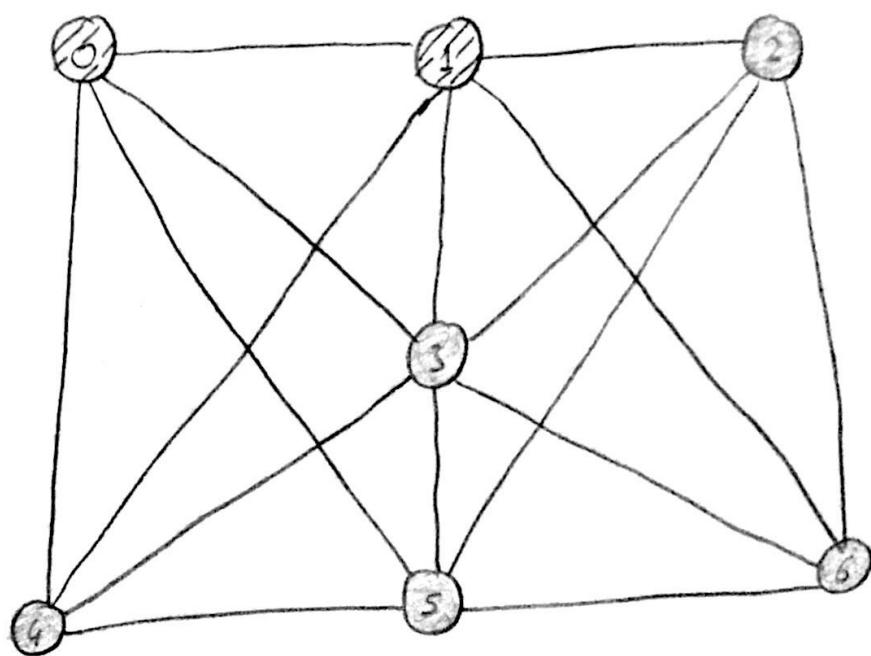
DVO = {2, 6, 5, 4, 3, 1, 0}

FO = { }
}



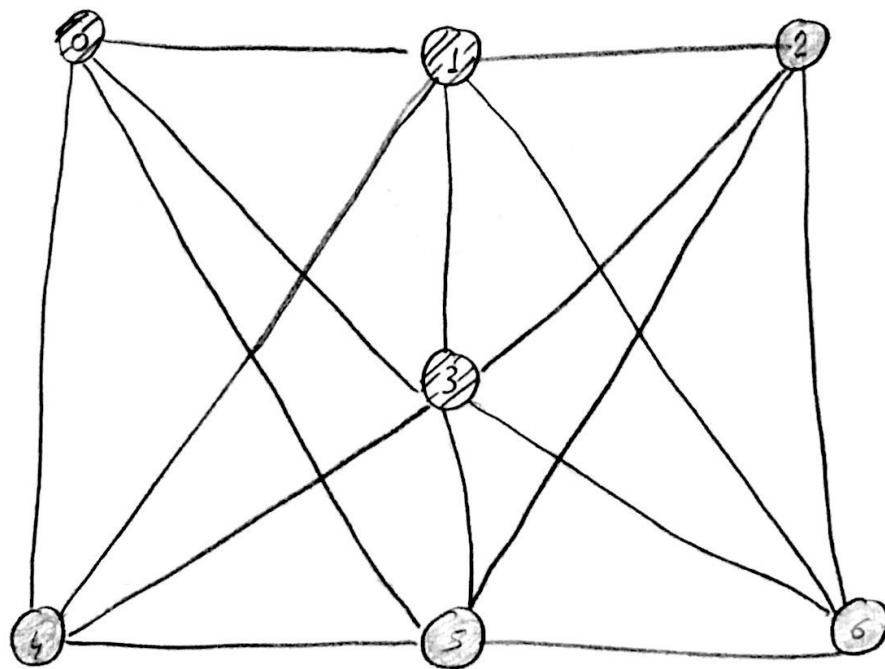
DVO = {2, 6, 5, 4, 3, 1, 0}

FO = {0}



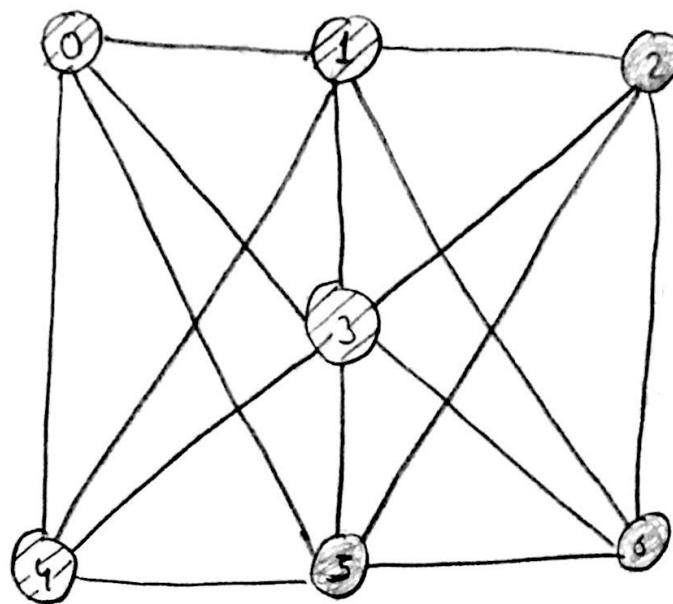
$DVO = \{2, 6, 5, 4, 3, 1; 0\}$

$FO = \{0, 1\}$

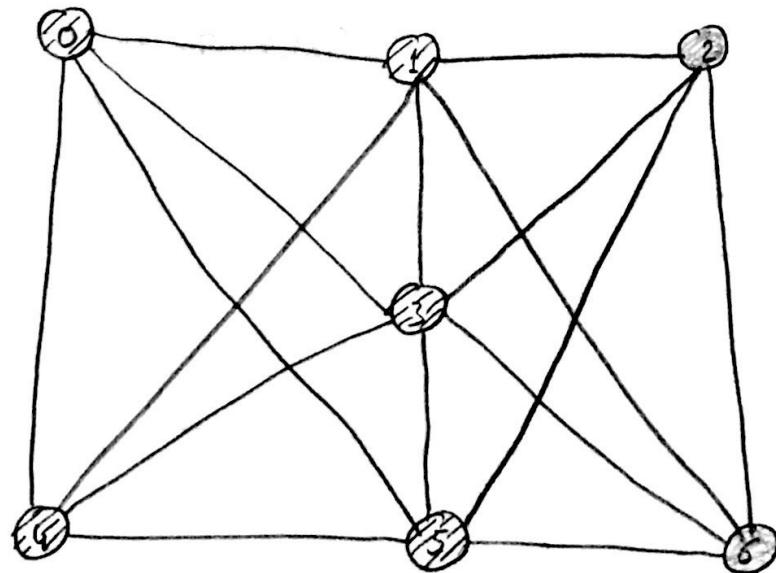


$DVO = \{2, 6, 5, 4, 3, 1; 0\}$

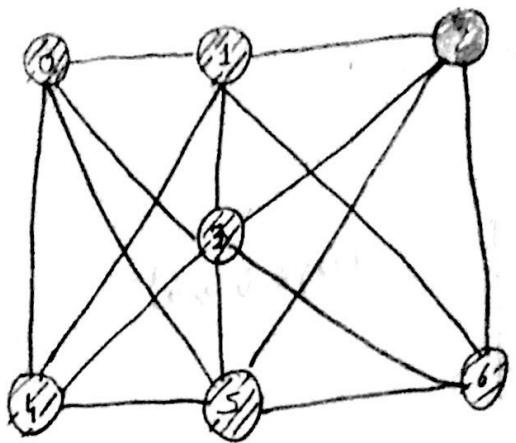
$FO = \{0, 1, 3\}$



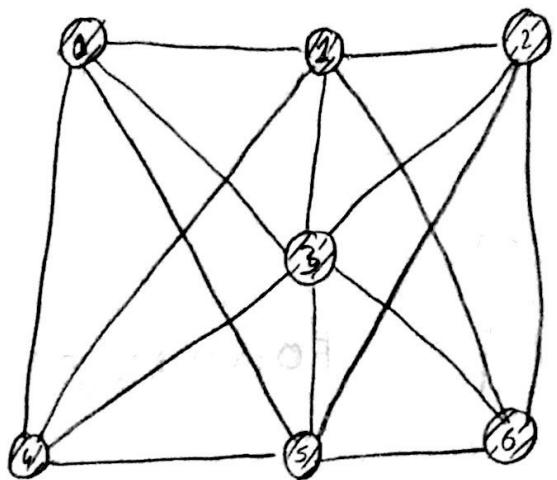
$$FO = \{0, 1, 3, 4\}$$



$$FO = \{0, 1, 3, 4, 5\}$$



$FO = \{0, 1, 3, 4, 5, 6, 2\}$

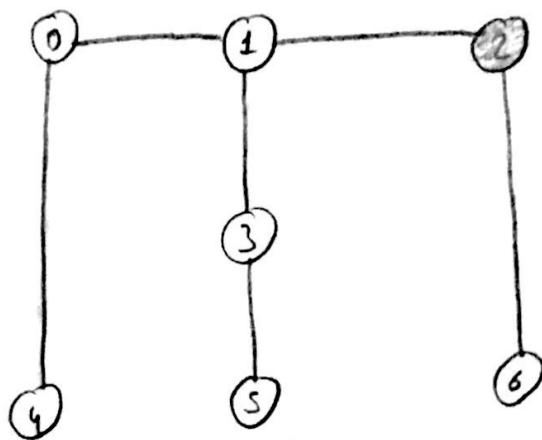


$FO = \{0, 1, 3, 4, 5, 6, 2\}$

Tree :

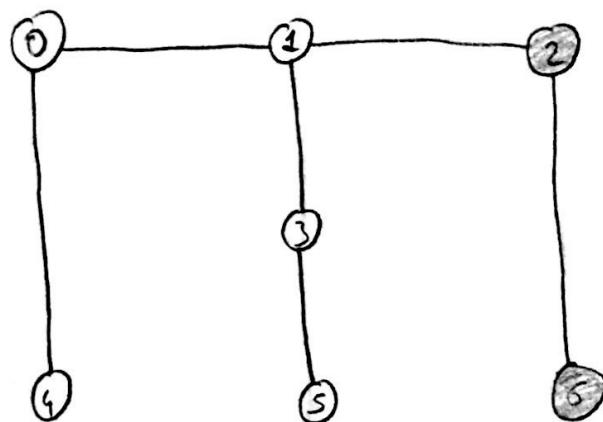


2nd Graph Deep First Search



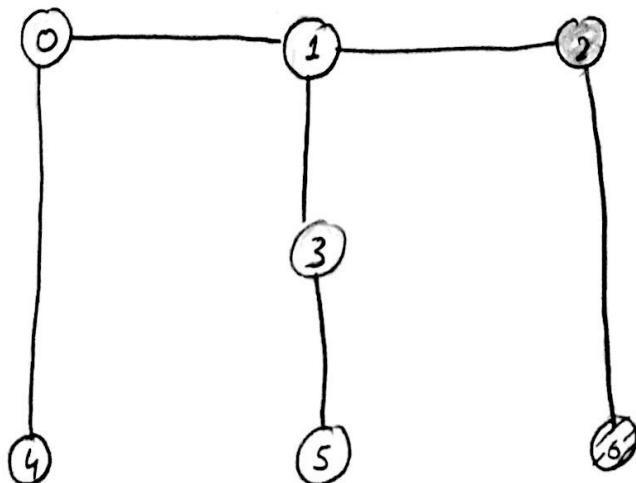
DVO = {2, 3}

FO = {0, 3}



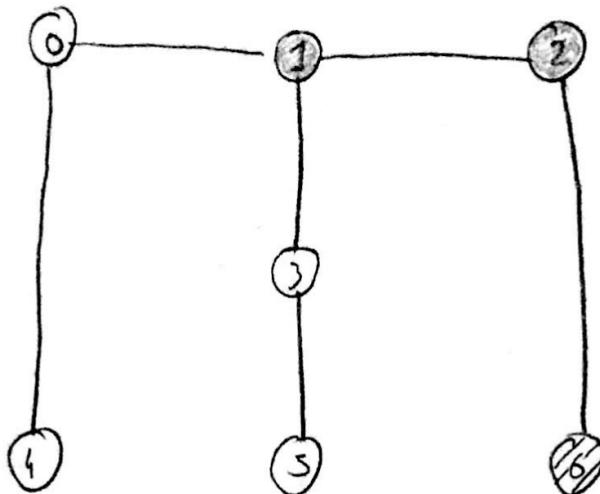
DVO = {2, 6}

FO = {3}



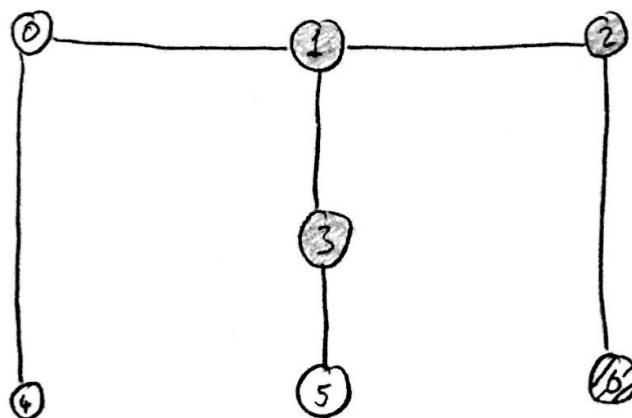
DVO = {2, 3}

FO = {6}



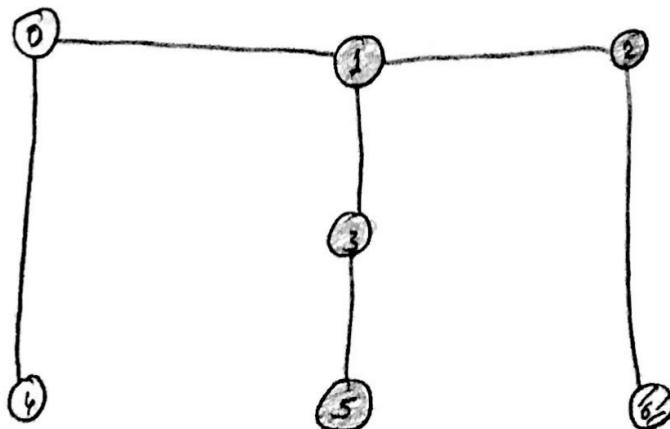
$$DVO = \{2, 6, 1\}$$

$$FO = \{6\}$$



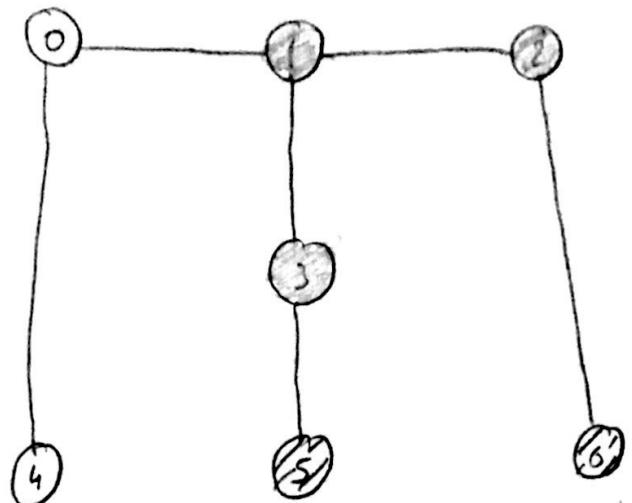
$$DVO = \{2, 6, 1, 3\}$$

$$FO = \{6\}$$



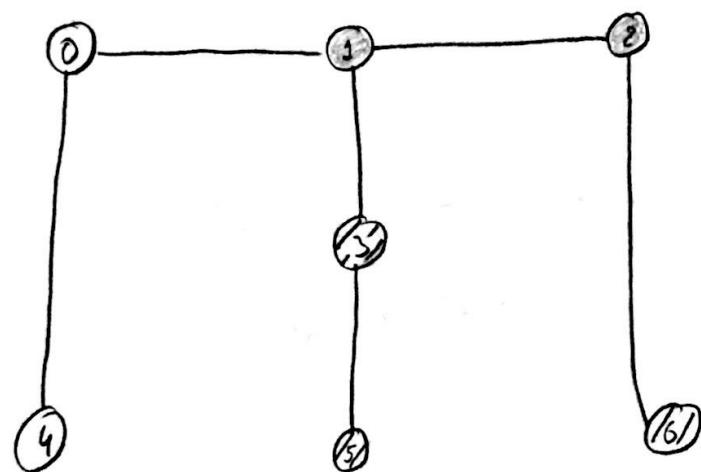
$$DVO = \{2, 6, 1, 3, 5\}$$

$$FO = \{6\}$$



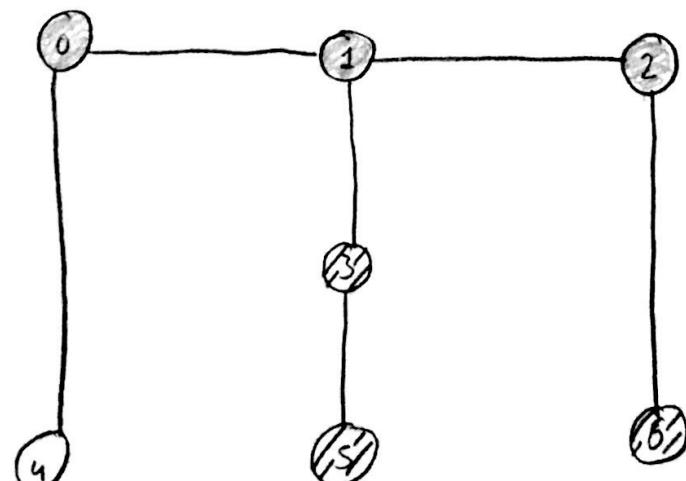
$$DVO = \{2, 6, 1, 3, 5\}$$

$$FO = \{6, 5\}$$



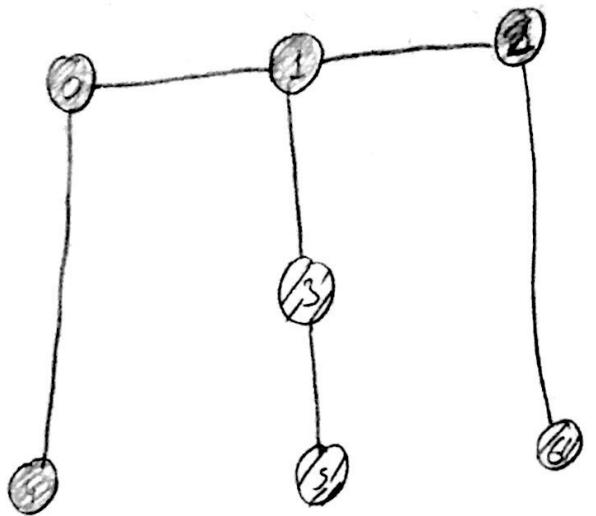
$$DVO = \{2, 6, 1, 3, 5\}$$

$$FO = \{6, 5, 3\}$$



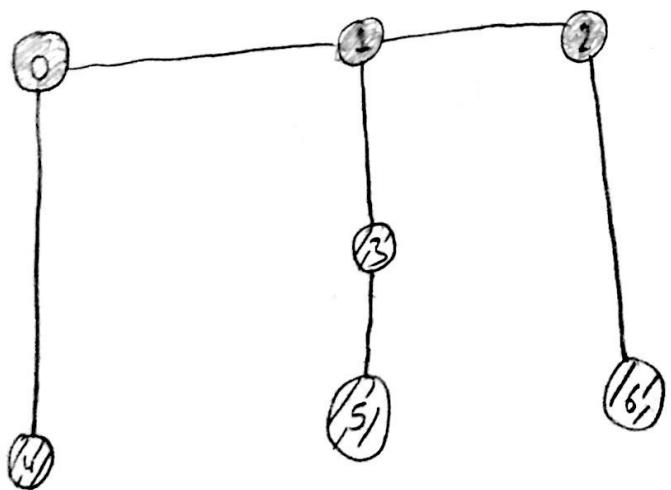
$$DVO = \{2, 6, 1, 3, 5, 0\}$$

$$FO = \{6, 5, 3\}$$



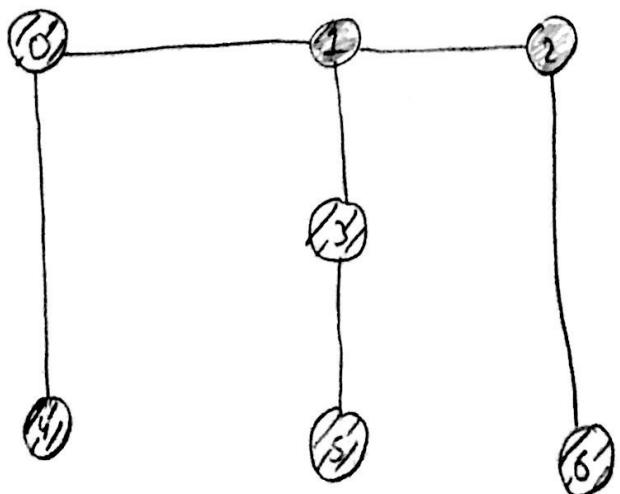
$$DVO = \{2, 6, 1, 3, 5, 0, 4\}$$

$$FO = \{6, 5, 2\}$$



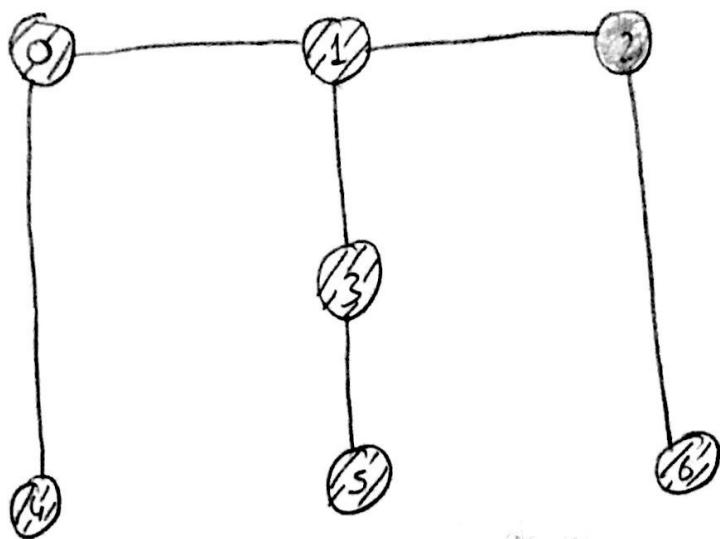
$$DVO = \{2, 6, 1, 3, 5, 0, 4\}$$

$$FO = \{6, 5, 2, 4\}$$



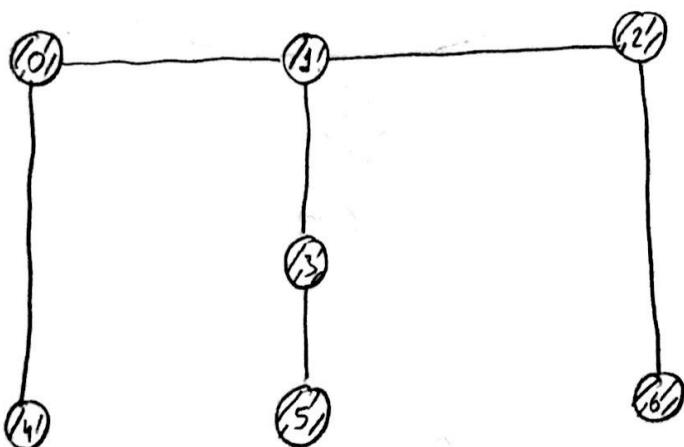
$$DVO = \{2, 6, 1, 3, 5, 0, 4\}$$

$$FO = \{6, 5, 2, 4, 0\}$$



$$DVO = \{2, 6, 1, 3, 5, 0, 4\}$$

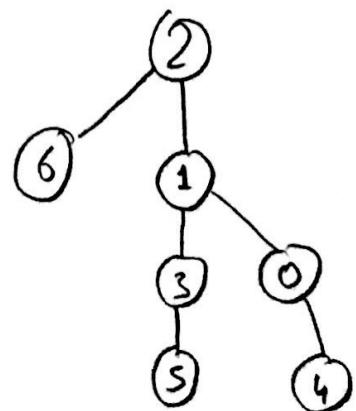
$$FO = \{6, 5, 3, 4, 0, 1\}$$



$$DVO = \{2, 6, 1, 3, 5, 0, 4\}$$

$$FO = \{6, 5, 3, 4, 0, 1, 2\}$$

Tree:



Analyze of Graph

Graph 1

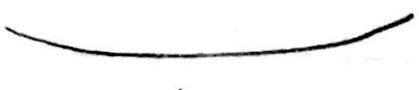
$$\frac{|E|}{|V|^2} = \frac{16}{(7)^2} = \frac{16}{49} \approx \frac{32}{100}$$



density

Graph 2

$$\frac{|E|}{|V|^2} = \frac{5}{(9)^2} = \frac{5}{81} \approx \frac{1}{10}$$



spars

- Depolama verimliliği için adjacency matrix kullanmak daha yararlıdır

- Depolama verimliliği için adjacency list kullanım daha yararlıdır. Çünkü $0.25^{'}$ in altında