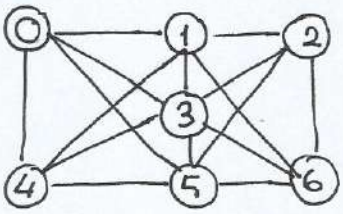
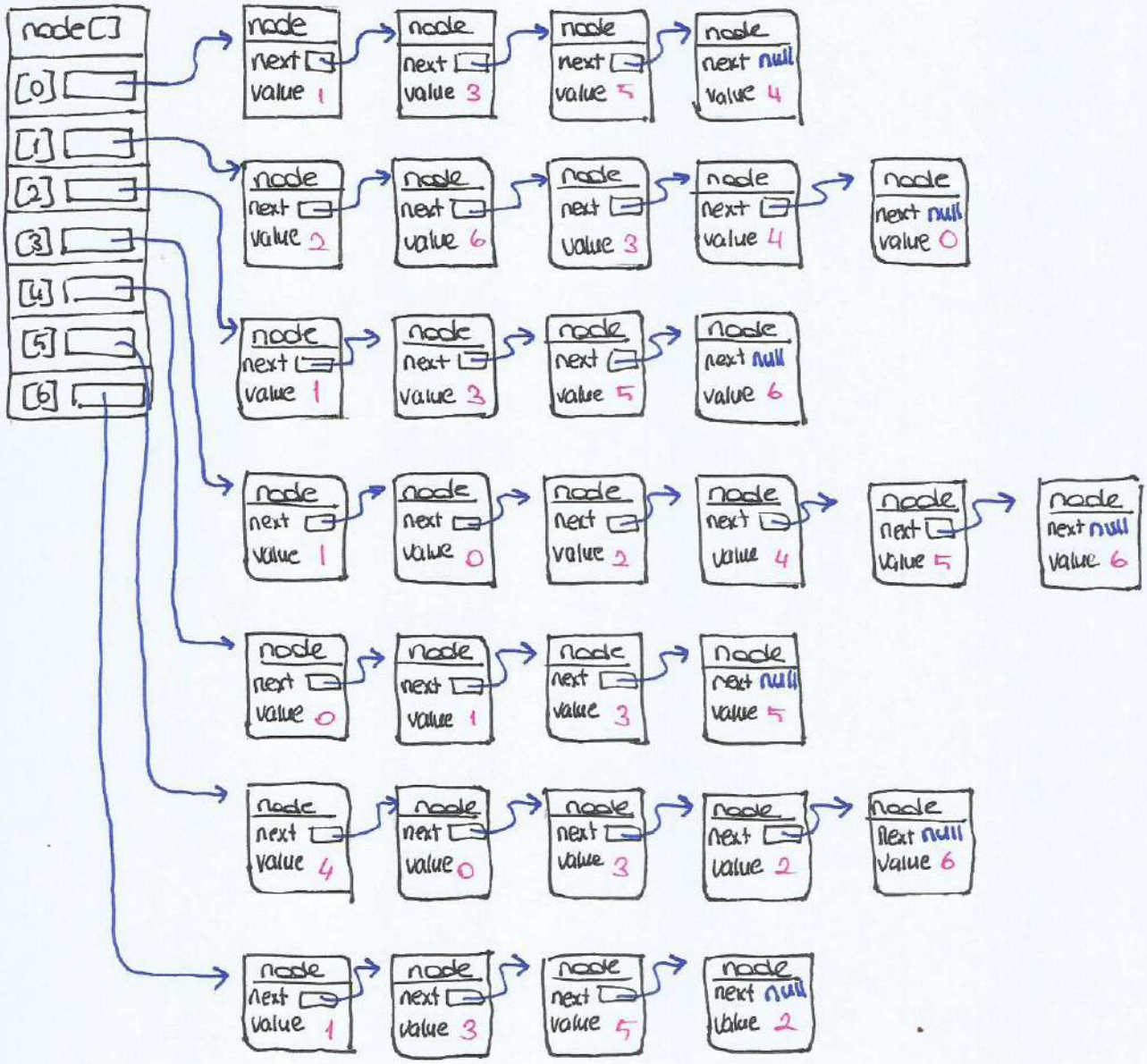
Elif Gonul
171044003

Q1)

1)

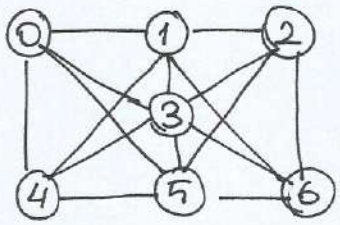


a) Represent the graph above using adjacency list.



Elif Gürel
171044003

1)

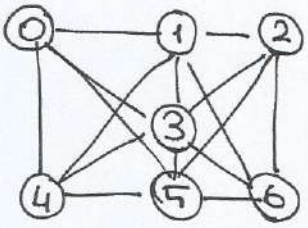


b) Represent the graph above using adjacency matrix.

	[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
[3]	1.0	1.0	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	

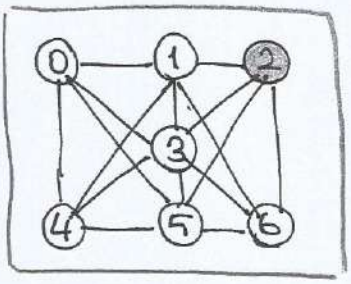
Elif Gorol
171044003

4)

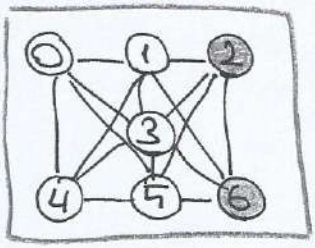


- DVO
- FO

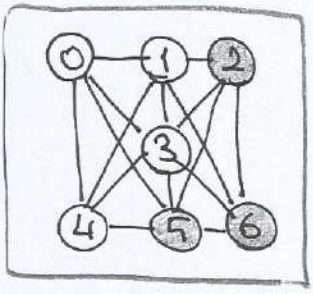
d) Draw DFS tree starting from vertex 2 and traversing the vertices adjacent to a vertex in descending order (largest to smallest)



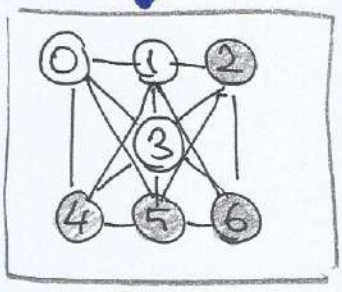
DVO={2}
FO={}



DVO={2,6}
FO={}

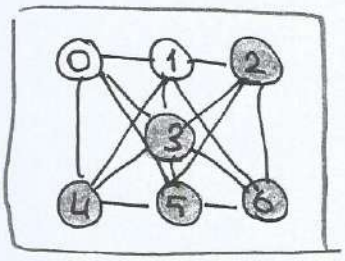


DVO={2,6,5}
FO={}

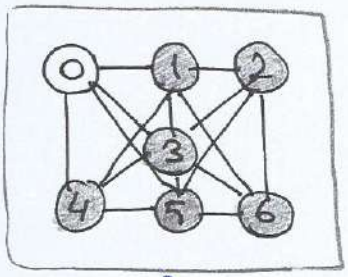


DVO={2,6,5,4}
FO={}

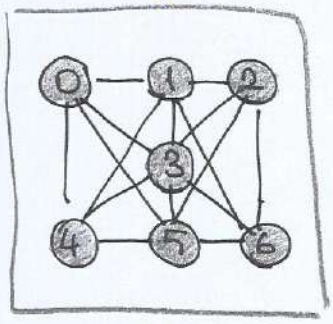
Elif Gool
171044003



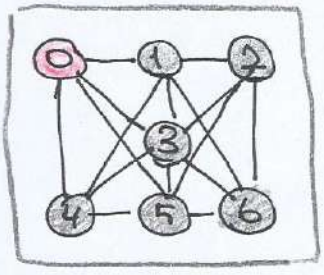
DVO= {2,6,5,4,3}
FO = {}



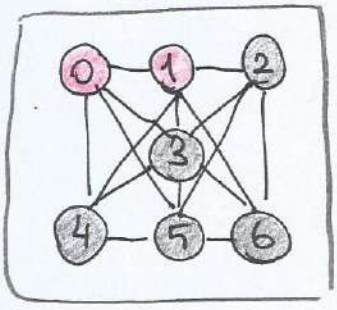
DVO = {2,6,5,4,3,1}
FO = {}



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

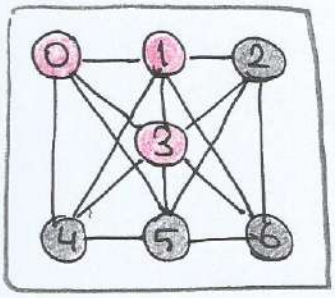


DVO = {2,6,5,4,3,1,0}
FO = {0}

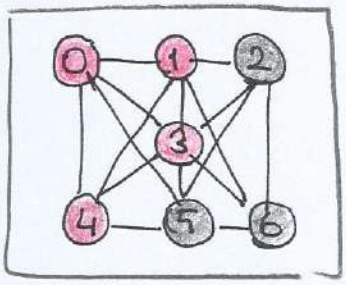


DVO = {2,6,5,4,3,1,0}
FO = {0,1}

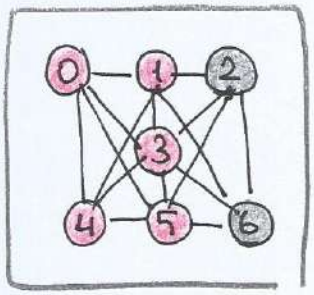
Elif Gonul
171044003



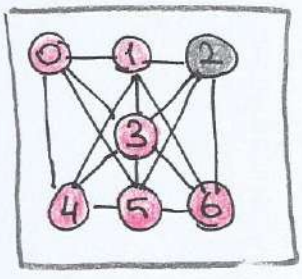
DVO = {2,6,5,4,3,1,0}
FO = {0,1,3}



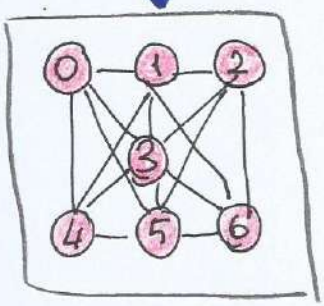
DVO = {2,6,5,4,3,1,0}
FO = {0,1,3,4}



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



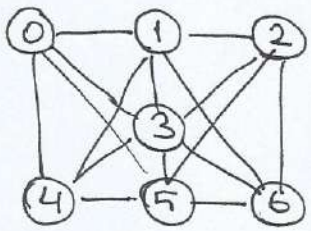
DVO = {2,6,5,4,3,1,0}
FO = {0,1,3,4,5,6}



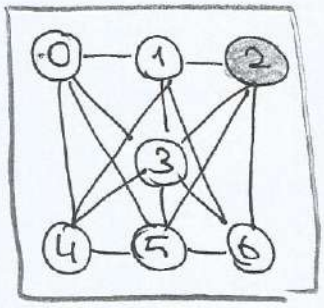
DVO = {2,6,5,4,3,1,0}
FO = {0,1,3,4,5,6,2}

Elif Gorol
171044003

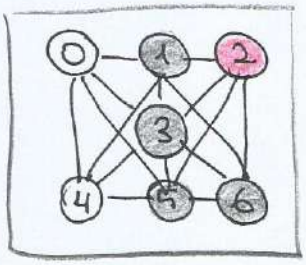
1)



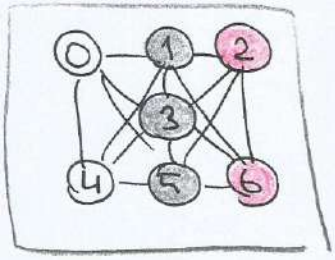
e) Draw BFS tree starting from a vertex 2 and traversing the vertices adjacent to a vertex in descending order (largest to smallest)



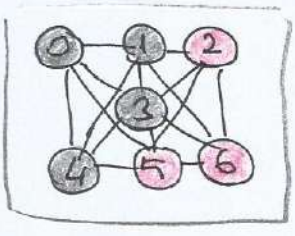
Queue = {2}
Visited = { }



Queue = {6,5,3,1}
Visited = {2}

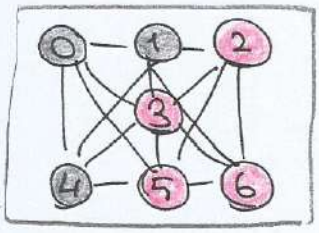


Queue = {5,3,1}
Visited = {2,6}

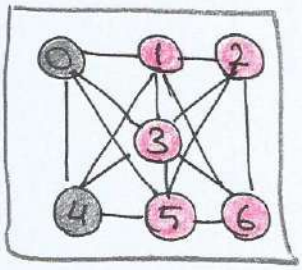


Queue = {3,1,4,0}
Visited = {2,6,5}

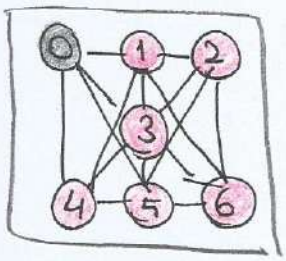
Elif Gonul
171044003



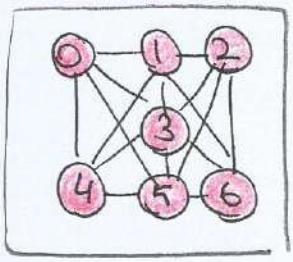
Queue = {1,4,0}
Visited = {2,6,5,3}



Queue = {4,0}
Visited = {2,6,5,3,1}



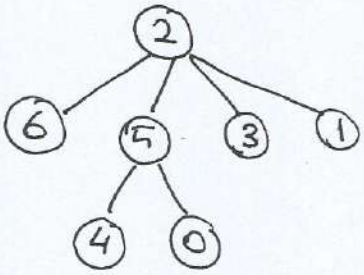
Queue = {0}
Visited = {2,6,5,3,1,4}



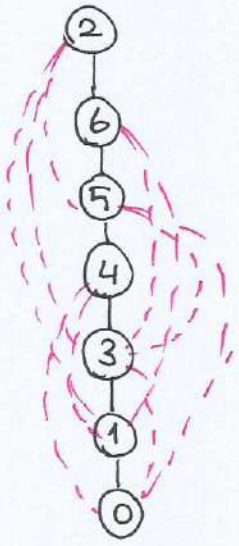
Queue = { }
Visited = {2,6,5,3,1,4,0}

Elif Genel
171044003

BFS tree of Graph 1

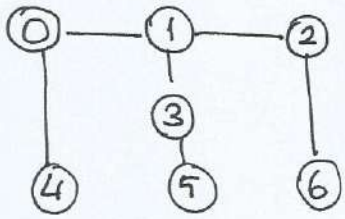


DFS tree of Graph 1

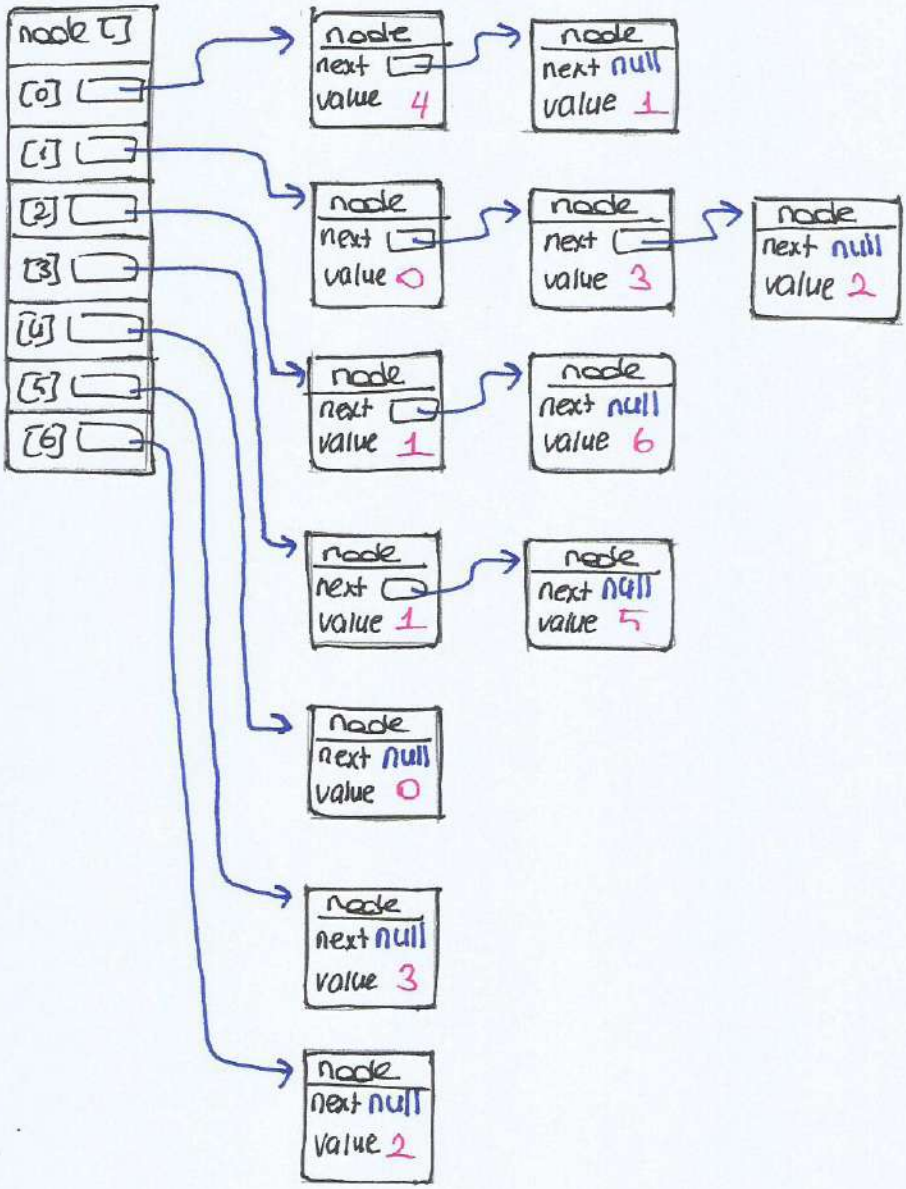


Elif Gorol
171044003

2)

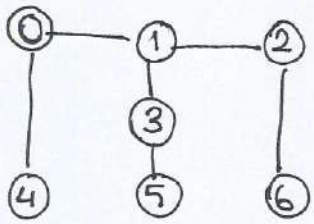


a) Represent the graph above using adjacency list.



Elif Gönül
171044003

2)

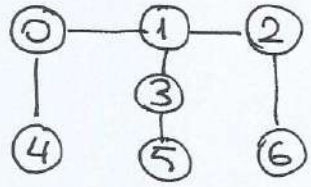


b) Represent the graph above using adjacency matrix

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

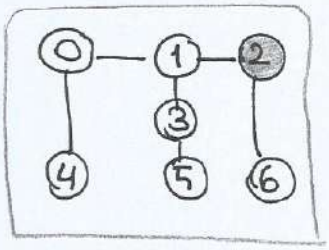
Elif Gürol
171044003

2)

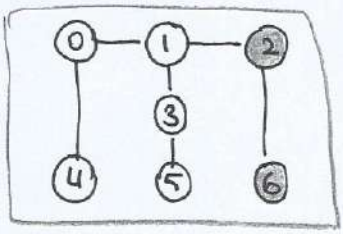


d) Draw DFS tree starting from vertex 1 and traversing the vertices adjacent to a vertex in descending order (largest to smallest)

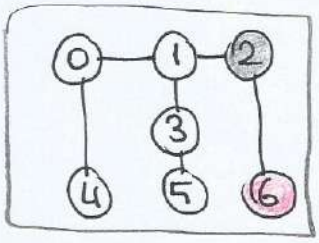
- DVO
- FO



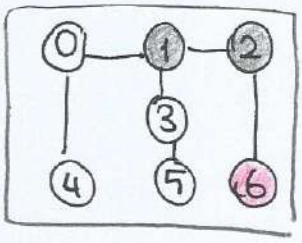
DVO = {2}
FO = { }



DVO = {2,6}
FO = 1 }

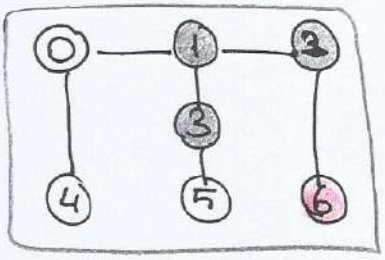


DVO = {2,6}
FO = {6}



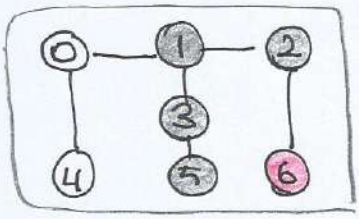
DVO = {2,6,1}
FO = {6}

Elif Gorel
171044003

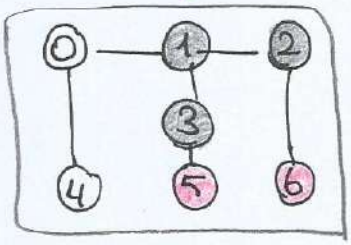


DVO = {2,6,1,3}

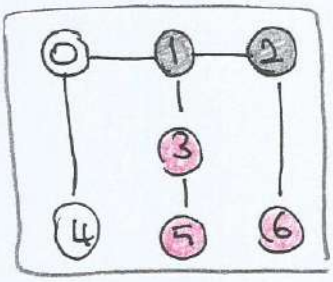
FO = {6}



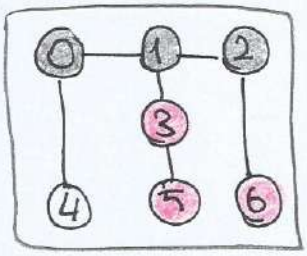
DVO = {2,6,1,3,5}
FO = {6}



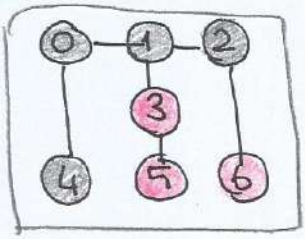
DVO = {2,6,1,3,5}
FO = {6,5}



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

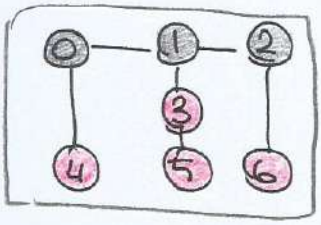


DVO = {2,6,1,3,5,0}
FO = {6,5,3}

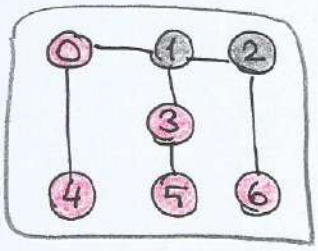


DVO = {2,6,1,3,5,0,4}
FO = {6,5,3}

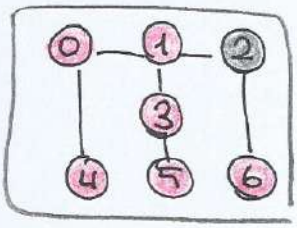
Elif Gorol
17010404003



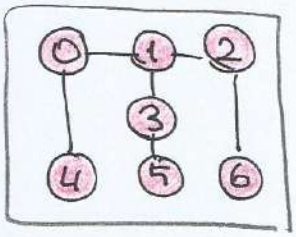
DVO = {2,6,1,3,5,0,4}
FO = {6,5,3,4}



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



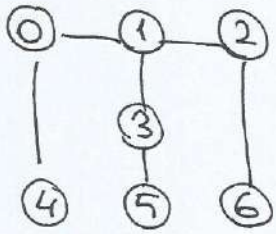
DVO = {2,6,1,3,5,0,4}
FO = {6,5,3,4,0,1}



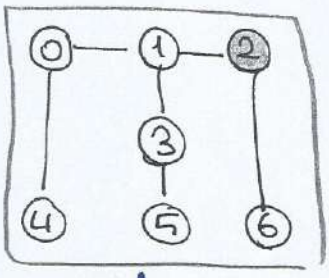
DVO = {2,6,1,3,5,0,4}
FO = {6,5,3,4,0,1,2}

Elif Gonul
171044003

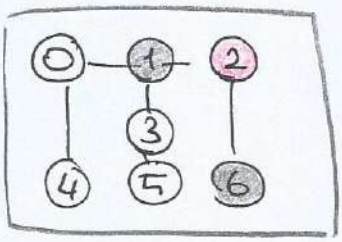
2)



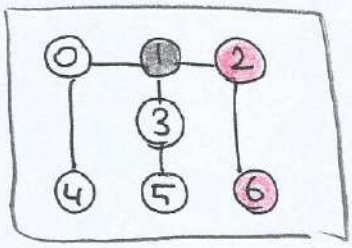
e) Draw BFS tree starting from vertex 2 and traversing the vertices adjacent to a vertex in descending order (largest to smallest)



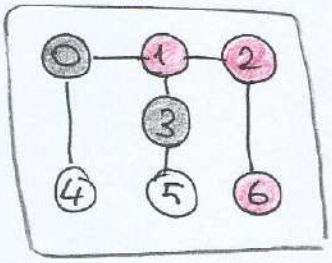
Queue = {2}
Visited = { }



Queue = {6,1}
Visited = {2}

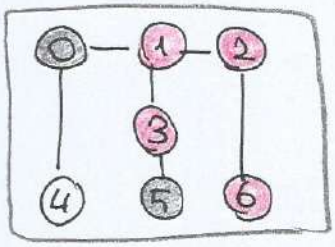


Queue = {1}
Visited = {2,6}

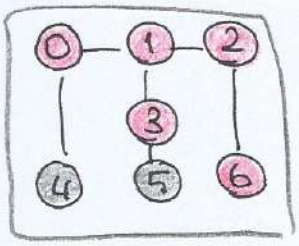


Queue = {3,0}
Visited = {2,6,1}

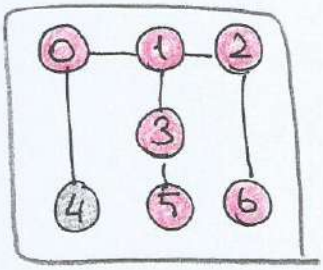
Elif Gonul
171044003



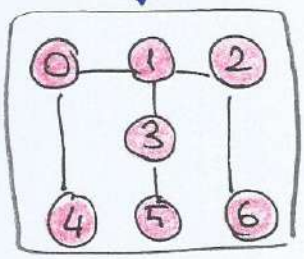
Queue = { 0,5 }
Visited = { 2,6,1,3 }



Queue = { 5,4 }
Visited = { 2,6,1,3,0 }



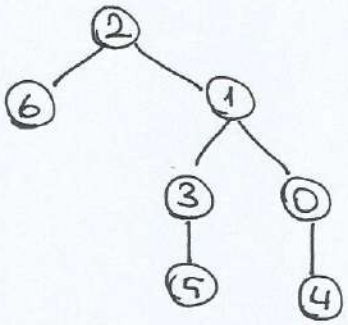
Queue = { 4 }
Visited = { 2,6,1,3,0,5 }



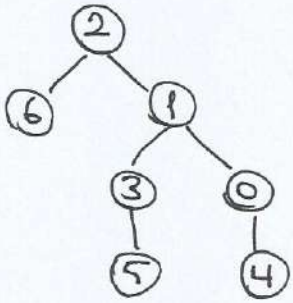
Queue = { }
Visited = { 2,6,1,3,0,5,4 }

Elif Gonul
170444003

BFS tree of Graph 2

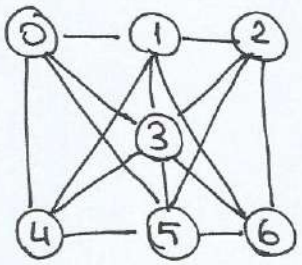


DFS tree of Graph 2



Elif Goral
171044003

1)

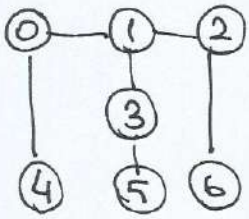


$|V| = n = 7$

$|E| = m = 16$

$$\text{density} = \frac{|E|}{|V|^2} = \frac{16}{49} = 0{,}326 \rightarrow \boxed{\text{dense graph}}$$

2)



$|V| = n = 7$

$|E| = m = 6$

$$\text{density} = \frac{|E|}{|V|^2} = \frac{6}{49} = 0{,}122 \rightarrow \boxed{\text{sparse graph}}$$

If graph has not many edges, there will be a lot of wasted space in the adjacency matrix. In adjacency list, only the adjacent edges are stored. On the other hand, in an adjacency list, each edge is represented by a reference to an Edge object containing data about the source, destination and weight. There is also a reference to next edge in a list. In a matrix representation, only the weight associated with an edge is stored. So each element in a adjacency list requires approximately four times the storage of an element in an adjacency matrix. So, the adjacency list uses less storage when less than 25 percent of the adjacency matrix would be filled. In our second graph, 12 of 49 space is full which means less than 25 percent of matrix is full. According to storage efficiency, second graph should be adjacency list. But in first graph, 32 of 49 space is full which means more than 25 percent of matrix is full. According to storage efficiency, first graph should be adjacency matrix.

Elif Gonul
171044093

$|E| < 0,25 . |V|^2 \rightarrow$ A.L.
$|E| > 0,25 . |V|^2 \rightarrow$ A.M.

1/ $|E| = 16$, $|V| = 7$ $\quad 16 \overset{?}{>} 0,25 . 49 \rightarrow \boxed{16 > 12,25} \rightarrow$ Adjacency matrix

2/ $|E| = 6$, $|V| = 7$ $\quad 6 \overset{?}{<} 0,25 . 49 \rightarrow \boxed{6 < 12,25} \rightarrow$ Adjacency list

According to time efficiency, if the graph is dense, the adjacency matrix representation is best, and if a graph is sparse, the adjacency list representation is best.

If we consider time efficiency and storage efficiency, adjacency matrix representation is better for first graph and adjacency list representation is better for second graph.