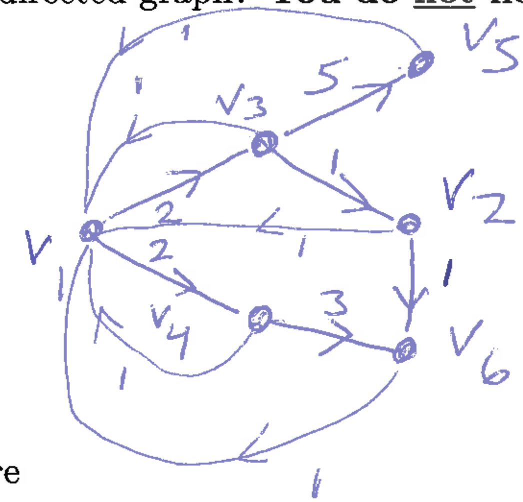
[8] 3.) Suppose a computer program uses the Breadth first search algorithm to determine which vertices are reachable from v_1 where the computer program gives lower indexed vertices priority (i.e., if the program must choose a vertex from a set of vertices, it will choose the one with lowest index). What would be the output if the input is the following adjacency matrix for a directed graph? **You do <u>not</u> need to show work.**

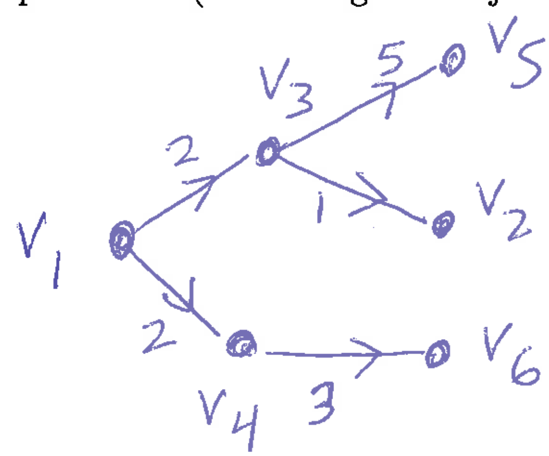
$$\begin{pmatrix} 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 5 & 0 \\ 1 & 0 & 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



The vertices reachable from v_1 are

$$v_1, v_2, v_3, v_4, v_5, v_6$$

Draw the tree created by the Breadth first search algorithm. Note this problem is related to problem 4 (same weighted adjacency matrix), but the ouput is not the same.



[5] 4a.) Define: A vertex w is reachable from a vertex v if

$$\exists \text{ a path from } v \text{ to } w$$

[15] 4b.) Suppose a computer program uses Dijkstra's algorithm to find a shortest path from the vertex v_1 to the vertex v_6 where the computer program gives lower indexed vertices priority (i.e., if the program must choose a vertex from a set of vertices, it will choose the one with lowest index). What would be the output if the input is the following adjacency matrix for a directed graph?

$$\begin{pmatrix} 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 5 & 0 \\ 1 & 0 & 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

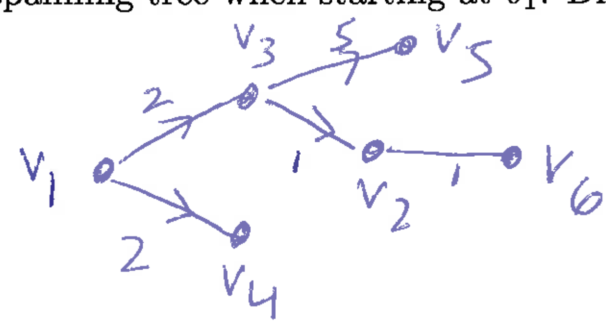
Show your work:

$S = \{v_1, v_3, v_4, v_2, v_6, v_5\}$

The table showing length of shortest paths found at each step:

v_1	v_2	v_3	v_4	v_5	v_6
0	∞	∞	∞	∞	∞
	∞	2	2	∞	∞
	3		2	7	∞
	3			7	5
				7	4
				7	

Note that every vertex is reachable from the vertex v_1. Thus Dikstra's algorithm outputs a spanning tree when starting at v_1. Draw this spanning tree:



What is a shortest path from the vertex v_1 to the vertex v_6?

$$v_1, < \overrightarrow{v_1, v_3} >, v_3, < \overrightarrow{v_3, v_2} >, v_2, < \overrightarrow{v_2, v_6} >, v_6$$

[5] 5a.) Define tournament:

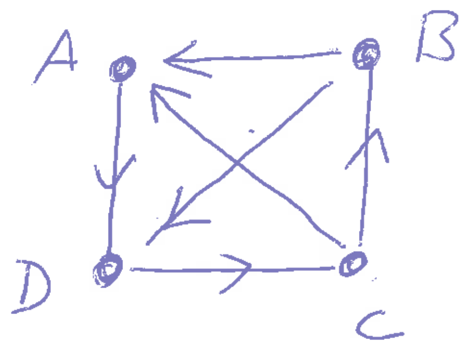
an orientation of a complete graph

I.e, a digraph D is a tournament if $\forall u, v \in V(D)$, exactly one of the arcs $< \overrightarrow{u, v} >$ or $< \overrightarrow{v, u} >$ is an arc in D.

[15] 5b.) The following is the result of a round robin tournament:

Team A beats Team D
Team B beats Team A
Team B beats Team D
Team C beats Team A
Team C beats Team B
Team D beats Team C

Draw the graph that models the above.



A Hamiltonian **path** in this graph is ADCB

Use this Hamiltonian path to assign 1st, 2nd, and 3rd prizes:

1st prize goes to team __*A*__. 2nd prize goes to team __*D*__. 3rd prize goes to team __*C*__.

A different Hamiltonian **path** in this graph is DCBA

1st prize goes to team __*D*__. 2nd prize goes to team __*C*__. 3rd prize goes to team __*B*__.

A different Hamiltonian **path** in this graph is BADC

1st prize goes to team __*B*__. 2nd prize goes to team __*A*__. 3rd prize goes to team __*D*__.

A different Hamiltonian **path** in this graph is BDCA

1st prize goes to team __*B*__. 2nd prize goes to team __*D*__. 3rd prize goes to team __*C*__.

A different Hamiltonian **path** in this graph is CBAD

1st prize goes to team __*C*__. 2nd prize goes to team __*B*__. 3rd prize goes to team __*A*__.