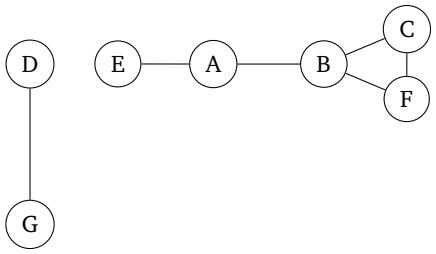
Section 06: Graphs

Section Problems

1. Graph properties

(a) Consider the *undirected, unweighted* graph below.



Answer the following questions about this graph:

(i) Find V, E, $|V|$, and $|E|$.

(ii) What is the maximum *degree* of the graph?

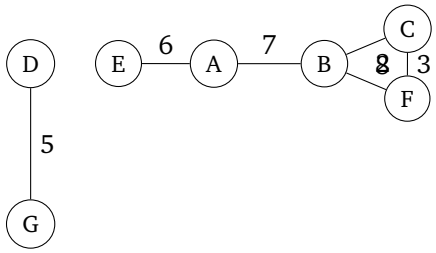
(iii) Are there any cycles? If so, where?

(iv) What is the maximum length simple path in this graph?

(v) What is one edge you could add to the graph that would increase the length of the maximum length simple path of the new graph to 6?

(vi) What are the *connected components* of the graph?

(b) Consider the *undirected, weighted* graph below.



Answer the following questions about this graph:

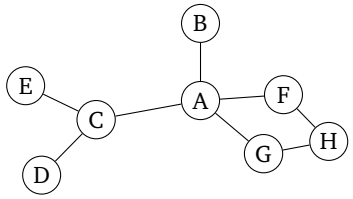
(i) What is the path involving the least number of nodes from E to C? What is its cost?

(ii) What is the minimum cost path from E to C? What is its cost?

(iii) What is the minimum length path from E to C? What is its length?

2. Graph traversal

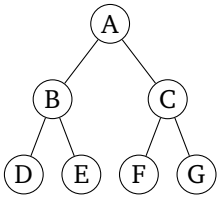
(a) Consider the following graph. Suppose we want to traverse it, starting at node A.



If we traverse this using *breadth-first search*, what are *two* possible orderings of the nodes we visit? What if we use *depth-first search*?

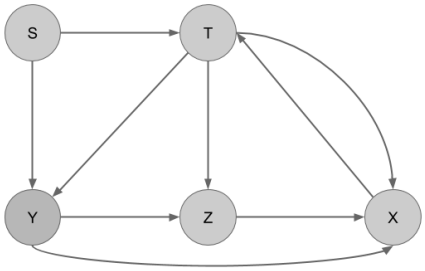
For the first ordering, you **must** run through adding/removing things from the queue/stack. To provide the second ordering for each algorithm, you may simply look at the graph.

(b) Same question, but on this graph:



3. Simulating BFS

Consider the following graph:



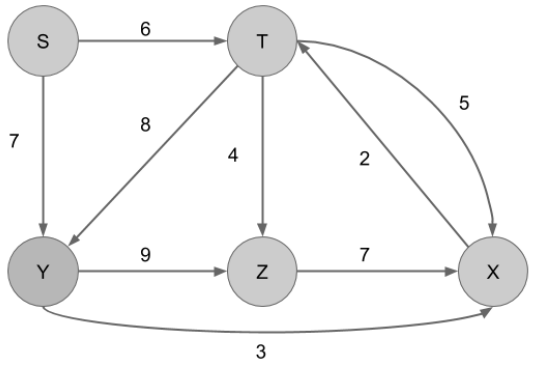
Run the BFS algorithm to find the shortest paths in this graph starting from vertex s. Draw out the queue of nodes, and also use the table below to keep track of each step in the algorithm. Finally, draw the resulting SPT (shortest path tree) after the algorithm has terminated.

Note: If two nodes enter the queue at the same time, break ties so that the node that comes first alphabetically exits the queue first.

Vertex	**Predecessor**	**Processed**
s		
t		
x		
y		
z		

4. Simulating Dijkstra's

(a) Consider the following graph:



Run Dijkstra's algorithm on this graph starting from vertex s. Use the table below to keep track of each step in the algorithm. Also draw the resulting SPT (shortest path tree) after the algorithm has terminated.

Vertex	**Distance**	**Predecessor**	**Processed**
s			
t			
x			
y			
z			

(b) Here is another graph. What are the final costs and resulting SPT (shortest path tree) if we run Dijkstra's starting on node A?