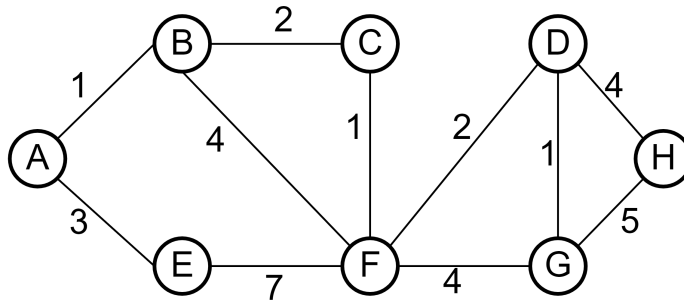


1 DFS, BFS, Dijkstra's, A*

For the following questions, use the graph below and assume that we break ties by visiting lexicographically earlier nodes first.



- Give the depth first search preorder traversal starting from vertex A .
- Give the depth first search postorder traversal starting from vertex A .
- Give the breadth first search traversal starting from vertex A .
- Give the order in which Dijkstra's Algorithm would visit each vertex, starting from vertex A . Sketch the resulting shortest paths tree.
- Give the path A* search would return, starting from A and with G as a goal.

Let $h(u, v)$ be the value returned by the heuristic for nodes u and v .

u	v	$h(u, v)$
A	G	9
B	G	7
C	G	4
D	G	1
E	G	10
F	G	3
H	G	5

2 Conceptual Shortest Paths

Answer the following questions regarding shortest path algorithms for a **weighted, undirected graph**. If the statement is true, provide an explanation. If the statement is false, provide a counterexample.

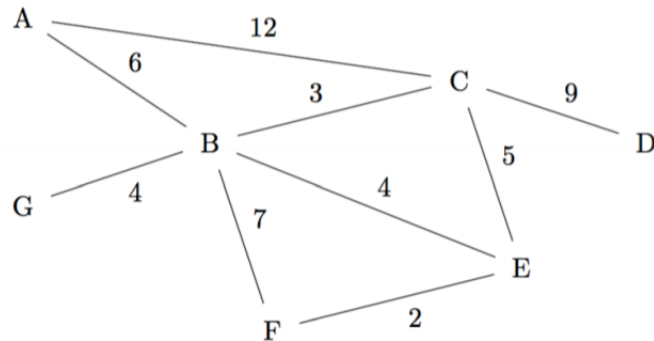
- (a) (T/F) If all edge weights are equal and positive, the breadth-first search starting from node A will return the shortest path from a node A to a target node B.

- (b) (T/F) If all edges have distinct weights, the shortest path between any two vertices is unique.

- (c) (T/F) **Adding** a constant positive integer k to all edge weights will not affect any shortest path between two vertices.

- (d) (T/F) **Multiplying** a constant positive integer k to all edge weights will not affect any shortest path between two vertices.

3 Introduction to MSTs



- (a) For the graph above, list the edges in the order they're added to the MST by Kruskal's and Prim's algorithm. Assume Prim's algorithm starts at vertex A. Assume ties are broken in alphabetical order. Denote each edge as a pair of vertices (e.g. AB is the edge from A to B)

Prim's algorithm order:

Kruskal's algorithm order:

- (b) Is there any vertex for which the shortest paths tree from that vertex is the same as your Prim MST? If there are multiple viable vertices, list all.
- (c) True/False: Adding 1 to the smallest edge of a graph G with unique edge weights must change the total weight of its MST
- (d) True/False: The shortest path from vertex A to vertex B in a graph G is the same as the shortest path from A to B using only edges in T , where T is the MST of G .
- (e) True/False: Given any cut, the maximum-weight crossing edge is in the maximum spanning tree.