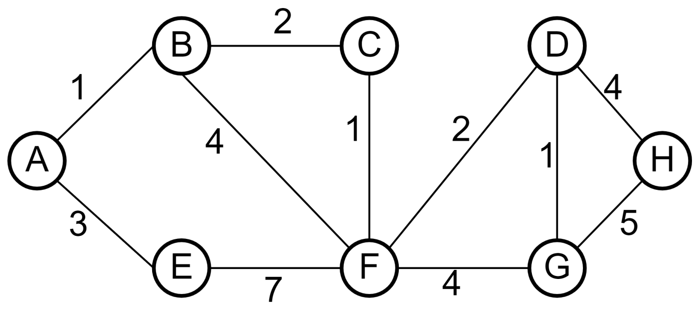
CS 61B
Spring 2021

Shortest Paths and MSTs

Exam Prep Discussion 10: March 29, 2021

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.



(a) Give the depth first search preorder traversal starting from vertex A.

A, B, C, F, D, G, H, E

(b) Give the depth first search postorder traversal starting from vertex A.

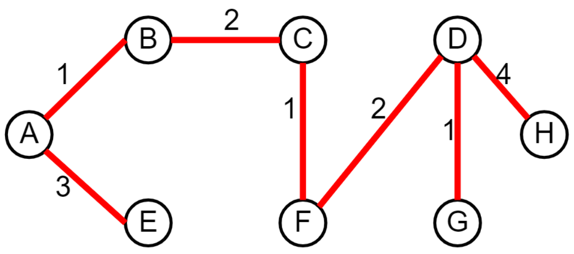
H, G, D, E, F, C, B, A

(c) Give the breadth first search traversal starting from vertex A.

A, B, E, C, F, D, G, H

(d) Give the order in which Dijkstra's Algorithm would visit each vertex, starting from vertex A. Sketch the resulting shortest paths tree.

A, B, C, E, F, D, G, H



(e) Give the path A* search would return, starting from A and with G as a goal. Let $h(u, v)$ be the valued 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

$A \rightarrow B, B \rightarrow C, C \rightarrow F, F \rightarrow D, D \rightarrow G$

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.

True. If all edges are equal in weight, then the shortest path from A to each node is proportional to the number of nodes on the path, so breadth first search will return the shortest path.

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

False. Consider a case of 3 nodes where AB is 3, AC is 5, and BC is 2. Here, the two possible paths from A to C both are of length 5. In general, paths with greater number of edges end up getting penalized more than paths with fewer edges.

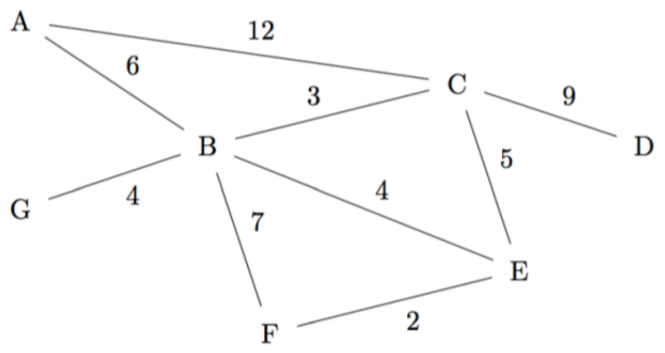
(c) (T/F) **Adding** a constant positive integer k to all edge weights will not affect any shortest path between two vertices.

False. Consider a case of 3 nodes A, B, and C where AB is 1, AC is 2.5 and BC is 1. Clearly, the best path from A to C is through B, with weight 2. However, if we add 1 to each edge weight, suddenly the path going through B will have weight 4, while the direct path is only 3.5.

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

True. Suppose we have arbitrary nodes u and v. Let's say the shortest path from u to v, before the multiplication by k, was of total weight w. This implies that every other path from u to v was of total weight greater than w. After multiplying each edge weight by k, the total weight of the shortest path becomes $w * k$ and the total weight of every other path becomes some number greater than $w * k$. Therefore, the original shortest path doesn't change.

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:

Prim's algorithm order: AB, BC, BE, EF, BG, CD
Kruskal's algorithm order: EF, BC, BE, BG, AB, CD

(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.

Vertex B, A, or G

(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

True, either this smallest edge (now with weight +1) is included, or this smallest edge is not included and some larger edge takes its place since there was no other edge of equal weight. Either way, the total weight increases.

(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.

False, consider vertices C and E in the graph above

(e) True/False: Given any cut, the maximum-weight crossing edge is in the maximum spanning tree.

True, we can use the cut-property proof as seen in class, but replace ”smallest” with ”largest”.