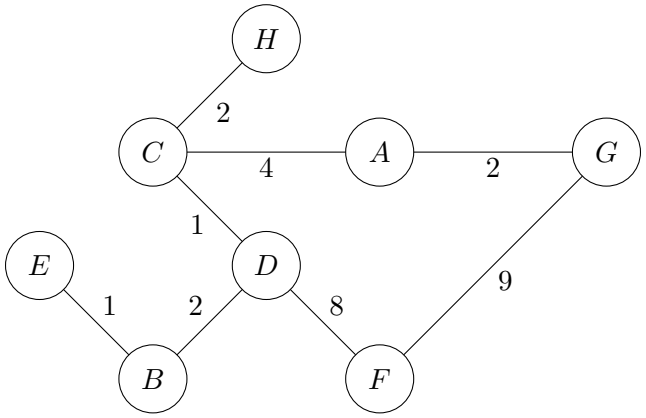
CSM 61B Fall 2020

Graph Algorithms

Mentoring 10: October 24-30, 2020

1 Searches

1.1 For the graph below, write the order in which vertices are visited using the specified algorithm starting from A. Break ties by alphabetical order. Notice that we have now introduced edge weights to the graph.



(a) DFS

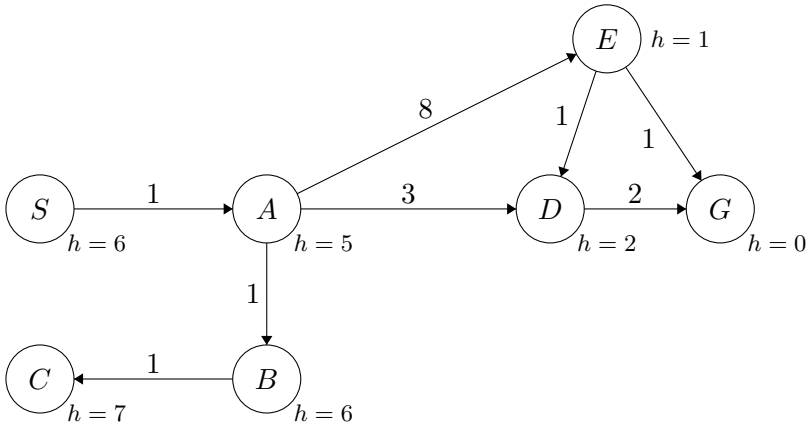
(b) BFS

(c) Dijkstra's

2 Shortest Paths

2.1 Find the path from the start, S, to the goal, G, when running each of the following algorithms.

The **heuristic**, h, estimates the distance from each node to the goal.



(a) Which path does Dijkstra's return?

(b) Which path does A* search return?

A* search is an algorithm that combines the total distance from the start with the heuristic to optimize the search procedure.

(c) What is the runtime of Dijkstra's? A*? What is the space requirement for both?

3 True and False

3.1 State if the following statements are True or False, and justify. For all graphs, assume that edge weights are positive and distinct, unless otherwise stated.

(a) Adding some positive constant k to every edge weight does not change the shortest path tree from vertex S.

(b) Doubling every edge weight does not change the shortest path tree.

(c) If the weight of each edge is decreased by 1, then the resulting shortest path in any graph from u to v is unchanged.

(d) If an edge e is the lightest edge connected to vertex S, it must be a part of the shortest path tree from vertex S.

(e) Consider a graph G, where every edge is nonnegative, except the edges adjacent to vertex s. Dijkstra's usually fails on graphs with negative edge weights, however if we run Dijkstra's starting from s, we will get the correct shortest paths tree.