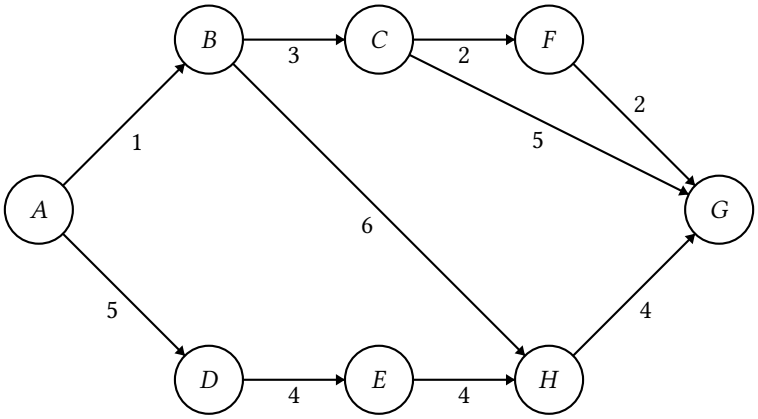
CSE 373 QuickCheck 5 Name: Evil Kevin Student ID: 1234567890



1. Give the order in which Dijkstra's Algorithm would visit each vertex starting from vertex A, where "visiting a vertex v" means "relaxing all of the edges out of v."

A B C D F H G E

For Djikstra's, we first insert all nodes in our fringe PQ, ordered by their distances from the source. Initially, these values are set to infinity, except the source, whose value is set to 0. We continually remove the smallest-valued vertex from the PQ and reset values of nodes in the PQ accordingly.

Start The fringe is initialized with the following values.

$$[A(0), B(\infty), C(\infty), D(\infty), E(\infty), F(\infty), G(\infty), H(\infty)]$$

A Visit A. Lower cost paths found for B and D.

$$[B(1), D(5), C(\infty), E(\infty), F(\infty), G(\infty), H(\infty)]$$

B Visit B. Lower cost paths found for C and H.

$$[C(4), D(5), H(7), E(\infty), F(\infty), G(\infty)]$$

C Visit C. Lower cost paths found for F and G.

$$[D(5), F(6), H(7), G(9), E(\infty)]$$

D Visit D. Lower cost path found for E.

$$[F(6), H(7), E(9), G(9)]$$

F Visit F. Lower cost path found for G.

$$[H(7), G(8), E(9)]$$

H Visit H. No lower cost path found.

$$[G(8), E(9)]$$

G Visit G. No outgoing neighbors.

$$[E(9)]$$

E No lower cost path found since H was already removed from the fringe.

$$[]$$

2. Change one of the weights in the graph so that the shortest paths tree returned by Dijkstra's is not correct. *Hint*: We showed in class that Dijkstra's shortest paths tree is correct so long as all edges are non-negative.

   Set the weight of the edge connecting vertex E and vertex H to the integer weight ≤ -3.

   To make the shortest paths tree incorrect, we need to find an edge in the graph where the node on the opposite end of the node being visited (i.e. the neighboring node) doesn't get considered because it had already been visited. We should be able to disregard the edge weight because we assume that once a vertex is visited, the shortest path to that node is finalized. However, this does not always hold true for graphs that have negative edge weights.

   (E, H) is the only edge for which this applies because H was removed from the fringe before E. Now that we know the weight of the edge is not considered, we need to find a value that will invalidate the shortest paths tree. Since we know (E, H) is the correct edge, we have to find shortest path that includes H. We see that (A, B, H) has a cost of 7, and (A, B, H, G) has a cost of 11. We know that (A, D, E) has a cost of 9, so we need to set the weight of (E, H) to a value such that (A, B, H) or (A, B, H, G) is no longer the shortest path to H or G. To do this, we can set (E, H) to any value ≤ -3 so that (A, D, E, H) has a cost ≤ 6, while (A, B, H) still has a cost of 7.

3. Suppose we use the following heuristic.

$$\begin{aligned} h(A, G) &= 2 \\ h(B, G) &= 2 \\ h(C, G) &= 20 \\ h(D, G) &= 2 \\ h(E, G) &= 6 \\ h(F, G) &= 2 \\ h(G, G) &= 0 \\ h(H, G) &= 2 \end{aligned}$$

   Recall that A* search is just Dijkstra's algorithm, except that the priority of a vertex v is given by the sum of the distance from the source to v plus $h(v, G)$, and also that we stop the search when the target is visited.

   Give the **path** (not order visited) that A* search returns from A to G. You may not need all blanks.

$A \quad B \quad H \quad \quad G$

We see $h(C, G) = 20$, but the actual cost to G is 4, so the heuristic overestimates the remaining distance from C to G.

Start The fringe is initialized with the following values.

$$[A(2), B(\infty), C, (\infty), D(\infty), E(\infty), F(\infty), G(\infty), H(\infty)]$$

A Smaller estimates found for B and D.

$$[B(3), D(7), C(\infty), E(\infty), F(\infty), G(\infty), H(\infty)]$$

B Smaller estimates found for C and H.

$$[D(7), H(9), C(24), E(\infty), F(\infty), G(\infty)]$$

D Smaller estimate found for E.

$$[H(9), E(15), C(24), F(\infty), G(\infty)]$$

H Smaller estimate found for G.

$$[G(11), E(15), C(24), F(\infty)]$$

G Visit G. Since G is the goal, we're done.