CS 101 Fall 2023 - Quiz 9

25/12/2023 - 20 Minutes **Name:** **ID number:**

---

1. (2 points) Notes of Discussion

I promise that I will complete this QUIZ independently and will not use any electronic products or paper-based materials during the QUIZ, nor will I communicate with other students during this QUIZ.

True or False: I have read and understood the notes. √ True ◯ False

2. (6 points) True or False

Note: You should write down your answers in the box below.

(a)	(b)	(c)
F	F	T

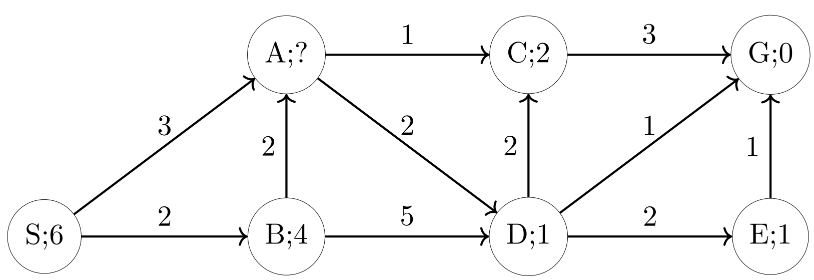
(a) (2') For any graph, if we run one additional iteration ($|V|$-th iteration) in Bellman-Ford algorithm, it will not make any change to `dist` array.

(b) (2') In A* graph search algorithm, if vertex u is marked visited before v, then `dist[u]` $\leq$ `dist[v]`, where `dist[u]` represents the real distance from start vertex to u.

(c) (2') Given an undirected graph $G = (V, E)$ with **positive integer** weights $\{w_e\}_{e \in E}$, we can modify the weights from w_e to $w_e + \frac{1}{|V|}$ for all edges $e \in E$. Then we can obtain a new graph G' such that the shortest paths in G' are the same as the shortest paths with the minimum number of edges in G.

3. (5 points) A* Algorithm

Suppose we are running A* graph search algorithm on the graph below, where S is the start vertex and G is the goal vertex. The heuristic function of each vertex is written inside the node.



(a) (2') Now $h(A)$ is unknown. For what value(s) of $h(A)$ will this graph be *consistent* and thus A* graph search will be guaranteed to return the optimal path? $h(A)$=______3______.

(b) (3') Choose and write down (one of) the possible heuristic you answered in (a). What is the order of the vertices being marked visited when we run A* graph search algorithm? Assume we break ties in alphabetical order, and we stop the algorithm once the goal is marked visited.

Solution: $h(A) = 3$. S,A,B,C,D,G

4. (7 points) Odd-Edge Shortest Path

Given a **directed** graph $G = (V, E)$, you need to find the shortest path from s to t that consists of an odd number of edges.

Input:

- $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, $E = \{(u_i, v_i, w_i > 0) : i = 1, 2, \ldots, m\}$.
- The start point s and the end point t.

Output:

- The edge weights sum of the shortest path from s to t that consists of an odd number of edges.
- If no such path return -1.

Note: In this question, you **don't** need to prove the correctness. You **need** to analyze the time complexity of your algorithm and ensure it **does not** exceed $O\left((|V| + |E|) \log |V|\right)$ to get full credits.

Hint: For this question, **don't** waste too much time trying to modify Dijkstra's Algorithm. Instead, try to create a new graph G' and run Dijkstra's Algorithm on the new graph G'.

Solution:

- In the new graph, duplicate all vertices v_i, represented by v_i^1 and v_i^2. (1 pts)
- For edge $e_i = (u_i, v_i, w_i)$, create edges (u_i^1, v_i^2, w_i) and (u_i^2, v_i^1, w_i) in the new graph. (2 pts)
- Run Dijkstra on the new graph from s^1 to t^2. If $d(s^1, t^2)$ equals to ∞, return -1, otherwise return $d(s^1, t^2)$. (2 pts)
- Duplicating vertices and creating new edges need $O(n + m)$ in total. G' has $2n$ points and $2m$ edges. Therefore, the time complexity of Dijkstra is $O((2n + 2m) \log n)$. The overall time complexity is $O((n + m) \log n) \sim O((|V| + |E|) \log |V|)$. (2 pts)