CS 101 Fall 2024 - Quiz 8

November 24, 2024 - 20 Minutes **Name:** **Student ID:**

---

1. **(2 points) Honor Code**

   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.

   I will not violate the Honor Code during this quiz.

   √ True ○ False

2. **(7 points) True or False**

   (a) (1') If we use BFS(breadth-first search) to find a path from u to v on an unweighted graph, this path will be the shortest path between u and v.

   √ True ○ False

   (b) (1') If we consider the disjoint set time complexity, the time complexity of Kruskal's algorithm can reach to $\Theta(|E|\alpha(|V|)\log|V|)$. ○ True √ False

   (c) (1') The maximum weight edge in a graph will not be in any MST. ○ True √ False

   (d) (1') Let G be a weighted undirected graph with positive weights where edge e has weight w_e for all $e \in E$. And G' is a copy of G except that every edge e has weight $w_e + \frac{2}{w_e}$. Then any MST in G is also a MST in G' (for every pair of (G, G')). ○ True √ False

   (e) (1') If T is a MST of G, then $\forall u, v \in G$ the path on the tree T connecting u and v is the shortest path from u to v in G. ○ True √ False

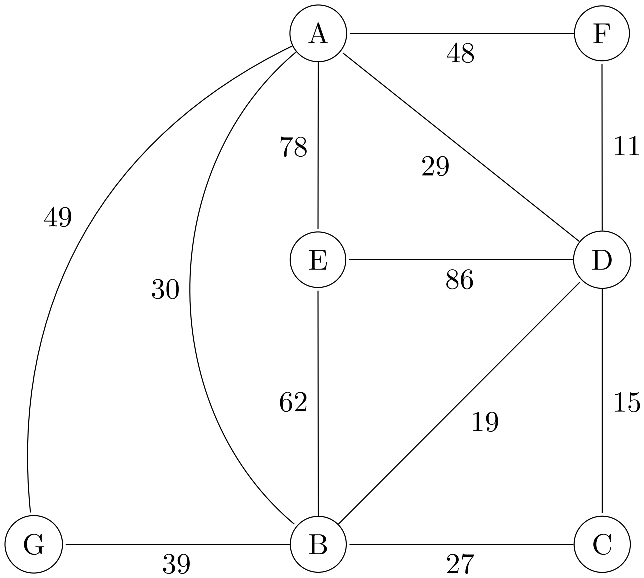
   (f) (1') Given a graph $G = (V, E)$, $w : E \to \mathbb{R}$ assigns the weight of every edge. $\forall C \subset E$ which is a cycle, if $\exists e$ satisfies $\forall e' \in C, w(e) > w(e')$, then e won't be in any MST of G. √ True ○ False

   > **Solution:**
   > Update: The True and False are all regarded as correct answer. The original intention of this problem is for the last sentence: $\exists e \in C$ satisfies $\forall e' \in C$.... Then this problem should choose True.

   (g) (1') Let $G = (V, E)$ be a connected undirected graph. If $e_0 \subset E$ is an edge such that $w(e_0) = \min\{w(e)|e \in E\}$, then e_0 belongs to every MST of G. ○ True √ False

3. **(6 points) Select the MST**

   In this problem, we want you to find the MST of the given graph. For every edge in the graph, check whether it is in the MST. **Select those edges which are in the MST in the table.**



Edge	Whether In MST
AB	◯ Selected
AD	√ Selected
AE	◯ Selected
AF	◯ Selected
AG	◯ Selected
BC	◯ Selected
BD	√ Selected
BE	√ Selected
BG	√ Selected
CD	√ Selected
DE	◯ Selected
DF	√ Selected

4. (10 points) MST with special edge

Given a weighted undirected graph $G = (V, E \cup \{\mathbf{e_0}\})$. For each edge in E, it can be represented as a triple: $e_i = (u_i, v_i, w_i)$. u_i and v_i mean the indices of the vertices connected by the edge, and w_i means the weight of the edge. There exists a special edge $\mathbf{e_0}$ which can be represented by $\mathbf{e}_0 = (u_0, v_0, w_0)$. We want you to find the minimal spanning tree containing the special edge $\mathbf{e}_0$ based on **Kruskal's algorithm**.

Please write the answers in the blanks below. Write the most precise and simplified form.

(You may use disjoint sets with union-by-rank and path-compression optimization in the problem. You can use 'Make-set', 'Union', 'Find-Set' to refer to the functions of those in the disjoint set. Take $\alpha(n)$ as a constant.)

Usage of three functions:

Make-Set **Require:** v vertex in V.
MAKE-SET(v) {Make a set in the disjoint sets.}

Find-Set **Require:** v vertex in V.
FIND-SET(v) {Find and return the corresponding set which v belongs to.}

Union **Require:** S_1, S_2 sets in the disjoint sets.
UNION(S_1, S_2) {Union S_1 and S_2 into a set in the disjoint sets.}

(A) Union(Find-Set(u_0),Find-Set(v_0)) or Union(Find-Set(v_0),Find-Set(u_0))

(B) $A \leftarrow \{\mathbf{e}_0\}$ or $A \leftarrow A \cup \{\mathbf{e}_0\}$

(C) Find-Set(u) != Find-Set(v) or Find-Set(v) != Find-Set(u)

(D) Union(Find-Set(u),Find-Set(v)) or Union(Find-Set(v),Find-Set(u))

- What's the time complexity of the algorithm? (Using $|E|$ and $|V|$) Θ($|E| \log |V|$ or $|E| \log |E|$)

Algorithm 1 Minimum Spanning Tree with Special Edge

Require: V vertex set, E edge set, $\mathbf{e}$ special edge

Ensure: Minimum Spanning Tree containing $\mathbf{e}_0$

$A \leftarrow \varnothing$
for each vertex v in set V **do**
    MAKE-SET(v)
end for
$(u_0, v_0, w_0) \leftarrow \mathbf{e_0}$
________ (A) ________
________ (B) ________
Sort(E , ascending) {Sort the edge in ascending weight}
for each edge e in set E **do**
    $(u, v, w) \leftarrow e$
    if ____ (C) ____ **then**
        $A \leftarrow A \cup \{e\}$
        ________ (D) ________
    end if
end for
return A