ShanghaiTech University

CS101 Algorithms and Data Structures
Fall 2024

Homework 8

Due date: November 27, 2024, at 23:59

1. Please write your solutions in English.
2. Submit your solutions to gradescope.com.
3. Set your FULL name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. `CamScanner` is recommended.
5. When submitting, match your solutions to the problems correctly.
6. No late submission will be accepted.
7. Violations to any of the above may result in zero points.

1. (12 points) Multiple Choices

Each question has **one or more** correct answer(s). Select all the correct answer(s). For each question, you will get 0 points if you select one or more wrong answers, but you will get 1 point if you select a non-empty subset of the correct answers.

Write your answers in the following table.

(a)	(b)	(c)	(d)	(e)	(f)
BD	AC	ABCD	ABD	A	AD

(a) (2') A planar graph is a graph which can be embedded in a plane i.e. you can find a way to put all vertices on the plane where the edges will not intersect with each other. Which of the statement(s) is/are correct?

A. $\forall n \leq 5, K_n$ is planar. K_n means the complete graph with n vertices.

B. K_6 is not planar.

C. DAGs are planar.

D. A tree is planar.

E. Bipartite graphs are planar.

(b) (2') Given a graph $G = (V, E)$, $w(e)$ indicates the weight of edge e. Which of the statement(s) is/are correct?

A. Both Kruskal's and Prim's algorithms can correctly find the MST even when $\exists e, w(e) < 0$.

B. Suppose G is connected and $|E| = \omega(|V|)$, G has a unique MST if and only if $\forall e, e' \in E, w(e) = w(e') \Leftrightarrow e = e'$ i.e. weights of edges are distinct.

C. Suppose $G' = (V, E)$ is the same graph as G with different weight function $v(e)$. If they share a same MST T, then T is also the MST of G with weights $u(e) = w(e) + v(e)$.

D. If G contains multi-edges i.e. G is not simple, then Kruskal's algorithm will fail but Prim's won't fail when finding MST.

(c) (2') Given a graph $G = (V, E)$, which of the following is(are) correct?

A. If G is a complete graph with 4 vertices, then the number of spanning trees of G is 16.

B. After Kruskal's algorithm, we choose m edges, then the number of connected components of G is $|V| - m$.

C. If G is stored in adjacency matrix, then the total time complexity of Kruskal's algorithm can reach $\Theta(|V|^2 + |E| \log |E|)$.

D. Suppose G is connected and $|V| = |E|$, the maximum number of spanning trees of G can reach $\Theta(|V|)$.

(d) (2') Let G be a weighted undirected graph with positive weights where edge e has weight $w_e \in \mathbb{R}^+$ for all $e \in E$. A new graph G', which is a copy of G, and the weight of each edge

e in G' is transformed using a function $f(w_e)$. Which of the following statements is/are true?

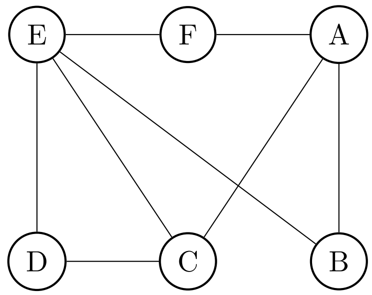
A. If $f(w_e) = w_e^2$, then any MST in G is also an MST in G'.

B. If $f(w_e) = 2^{w_e}$, then any MST in G is also an MST in G'.

C. If $f(w_e) = \frac{1}{w_e}$, then any MST in G is also an MST in G'.

D. If $f(w_e) = \log(w_e)$, then any MST in G is also an MST in G'.

(e) (2') What is the number of spanning trees of following graph?



A. 32

B. 34

C. 36

D. 38

(f) (2') Which of the following statements are true for MST(Minimum Spanning Tree)?

A. Suppose G has multiple MSTs. For each minimum spanning tree T of a graph G, there is a way to sort the edges of G in Kruskal's algorithm so that the algorithm returns T.

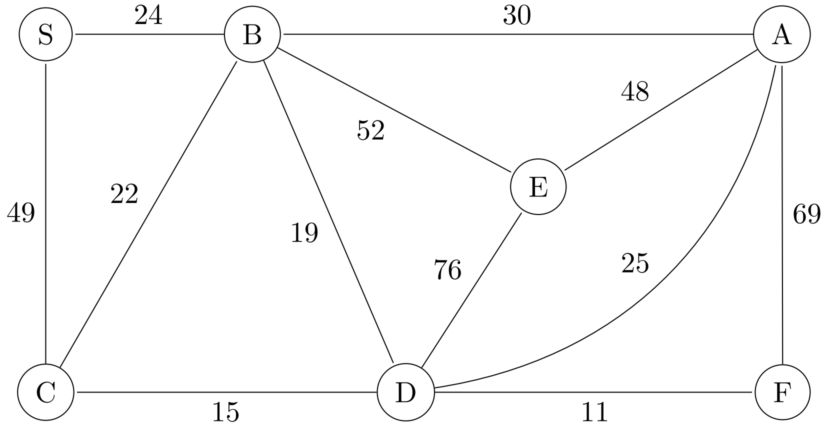
B. Prim's algorithm is a divide-and-conquer algorithm because it divides the graph into S and $V - S$ then solve.

C. If we use binary heap to optimize Prim's algorithm when choosing the next edge, it will always have a better time complexity than the original algorithm on any graph.

D. If we add a new edge $e = (u, v)$ into a graph $G = (V, E)$ with unique MST to get a new graph $G' = (V, E \cup \{e\})$. There is at most 1 edge difference between the MST of G and G'.

2. (20 points) Simulation of MST

Given a graph G as below:



(a) (6') Use Prim's algorithm to find the Minimal Spanning Tree of the graph. You should select S as the root node. Write the visit order of all nodes.

S→ **B** → **D** → **F** → **C** → **A** → **E**

(b) (6') Use Kruskal's algorithm to find the Minimal Spanning Tree of the graph. Write the edges chosen in order.

DF → **CD** → **BD** → **BS** → **AD** → **AE**

(c) (2') Are the MST obtained by Prim's algorithm and the one obtained by Kruskal's algorithm the same ? (**Please write "Yes" or "No".**) **Yes**

(d) (6') Let's modify the weights of some edges. Please give out the maximum and minimum weight of the edges given that won't change the MST. (You can write $+\infty$ or $-\infty$ if there is no maximum or minimum weight. You should consider every method of breaking ties.)

- AD: Maximum: 30 Minimum: $-\infty$
- BC: Maximum: $+\infty$ Minimum: 19
- DE: Maximum: $+\infty$ Minimum: 48

3. (8 points) Designing machine

Fritia is designing a new machine with n components and m wires, with each wires connecting two different components. You can consider this as a connected graph $G = (V, E)$. Denote $e_i \in E$ as $e_i = (u_i, v_i, s_i)$ where e_i connects u_i-th and v_i-th components and has a maximum transmission speed limit s_i.

To test her machine, she starts importing data into the 1st component. Unfortunately, each wire has a distinct maximum transmission speed limit s_i. Fritia wants to find a path which can transmit data as fast as possible for each component. **(The transmission speed limit of a path is the minimum of the maximum transmission speed limit of every wire.)**

Your task is $\forall 2 \leq i \leq |V|$, find a path from 1st component to i-th component, which has the fast transmission speed limit. You should give out the steps of your algorithm (**as efficient as possible**), brief reason of correctness together with the time complexity (tight).

Hint: Recall how Kruskal's works. You can use any algorithm taught in class directly.

Solution: Description: (5pt)

- Find the **maximum** spanning tree of G called T, with weight s_i.
- Traverse the tree T from 1, using DFS i.e. see T as a rooted tree with root 1 then use DFS to traverse it. And initialize the $i-$th answer as ans_i (see ans_1 as $+\infty$.)
- When it comes to the i-th component ($i \neq 1$), we can obtain its parent on the tree, denote it as par_i-th component. And the wire between them as $e = (par_i, i, s)$. Then the maximum transmission speed limit of $i-$th component is the minimum between the maximum transmission speed limit of par_i-th component and the maximum transmission speed limit of s i.e. $ans_i = \min\{ans_{par_i}, s\}$.

Correctness: (2pt)

- Use contradiction: For vertex i, if $\exists$ an edge e' not in MST and a path containing e' which has a higher limit. Add the edge into the maximum spanning tree to get T'. There must exists a cycle in T', and we know the cycle must contain some edge e'' with lower limit otherwise the path won't have a higher limit using e'. Then deleting e'' and adding e' will make a new spanning tree which has more weights, contradicts to MST. (2pts)
- G is connected so every vertices can be reached. (1pts, but not extra points.)

Time complexity: $O(|E| \log |E|)$. (MST takes $O(|E| \log |E|)$, the traverse takes $O(|V|)$.) (1pt). (The efficiency of your algorithm will also matter. If not efficient enough, you will lose points.)