COL351 Quiz 1

Viraj Agashe

TOTAL POINTS

6 / 10

QUESTION 1

1 Q1 3 / 3

✓ **+ 3 pts** **Correct Linear Time Algorithm**

+ 0 pts Incorrect / did not attempt

+ 1.5 pts Correct Polynomial(not Linear) Time Algorithm

+ 1.5 pts Linear Time Algorithm Partial

+ 1.5 pts Proof of Correctness

+ 1 pts Proof of Correctness Partial

+ 1 pts Proof of Time Complexity

+ 0.5 pts Checking if 0 degree vertices exist

+ 0.5 pts Exponential Time Algorithm

- 6 pts Cheating

+ 1 pts Point Adjustment

QUESTION 2

2 Q2 3 / 3

✓ **+ 3 pts** **Correct**

+ 0 pts Incorrect

QUESTION 3

3 Q3 0 / 4

✓ **+ 0 pts** **Incorrect**

+ 4 pts Correct

1 Q1 3 / 3

- ✓ **+ 3 pts** **Correct Linear Time Algorithm**
- **+ 0 pts** Incorrect / did not attempt
- **+ 1.5 pts** Correct Polynomial(not Linear) Time Algorithm
- **+ 1.5 pts** Linear Time Algorithm Partial
- **+ 1.5 pts** Proof of Correctness
- **+ 1 pts** Proof of Correctness Partial
- **+ 1 pts** Proof of Time Complexity
- **+ 0.5 pts** Checking if 0 degree vertices exist
- **+ 0.5 pts** Exponential Time Algorithm
- **- 6 pts** Cheating
- **+ 1 pts** Point Adjustment

2 Q2 3 / 3

✓ + 3 pts Correct

+ 0 pts Incorrect

3 Q3 0 / 4

✓ **+ 0 pts** **Incorrect**

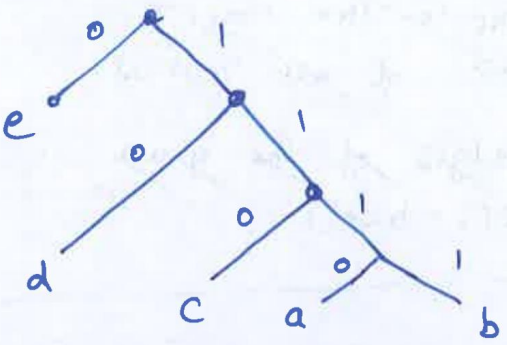
+ 4 pts Correct

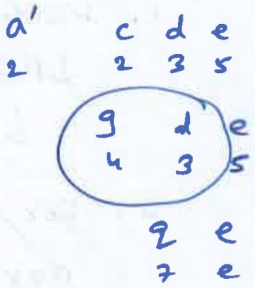
QUIZ 1.

(6)

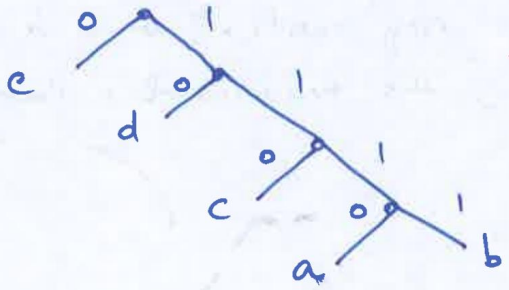
VIRAJ AGASHE
2020CS10567

1. {a, b, c, d, e} → 1, 1, 2, 3, 5





PREFIX TREE



+3

---

2. Note that there can be at most n connected components in a graph, i.e. n-1 bridge edges implies that removing each edge breaks the graph into 2 connected components, i.e. each bridge edge connects 2 vertices & is the only path between the 2 vertices.

∴ For a connected graph with n-1 bridge edges ~~each vertex~~ the graph must be acyclic

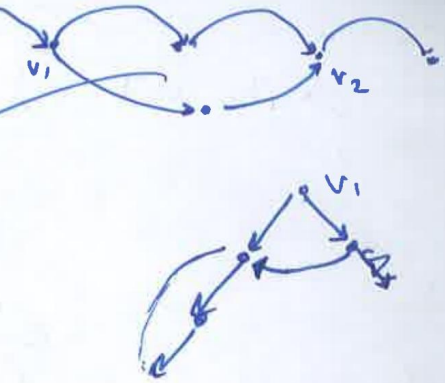
A connected & acyclic graph is a tree.

∴ G itself is a spanning tree of the graph.

i.e. the graph is a tree & is its own spanning tree (& therefore MST)

+3

3. $G = (V, E)$



Algorithm

1. Perform a DFS traversal of the DAG, and compute the start time & finish times of all vertices.
2. For all the edges of the graph not in the DFS tree :

---

3. $G = (V, E)$

Algorithm

1. Perform a DFS of the graph G.
2. If we encounter any vertex which is already visited during the traversal, then there is not a unique path

Time : $O(m+n)$
