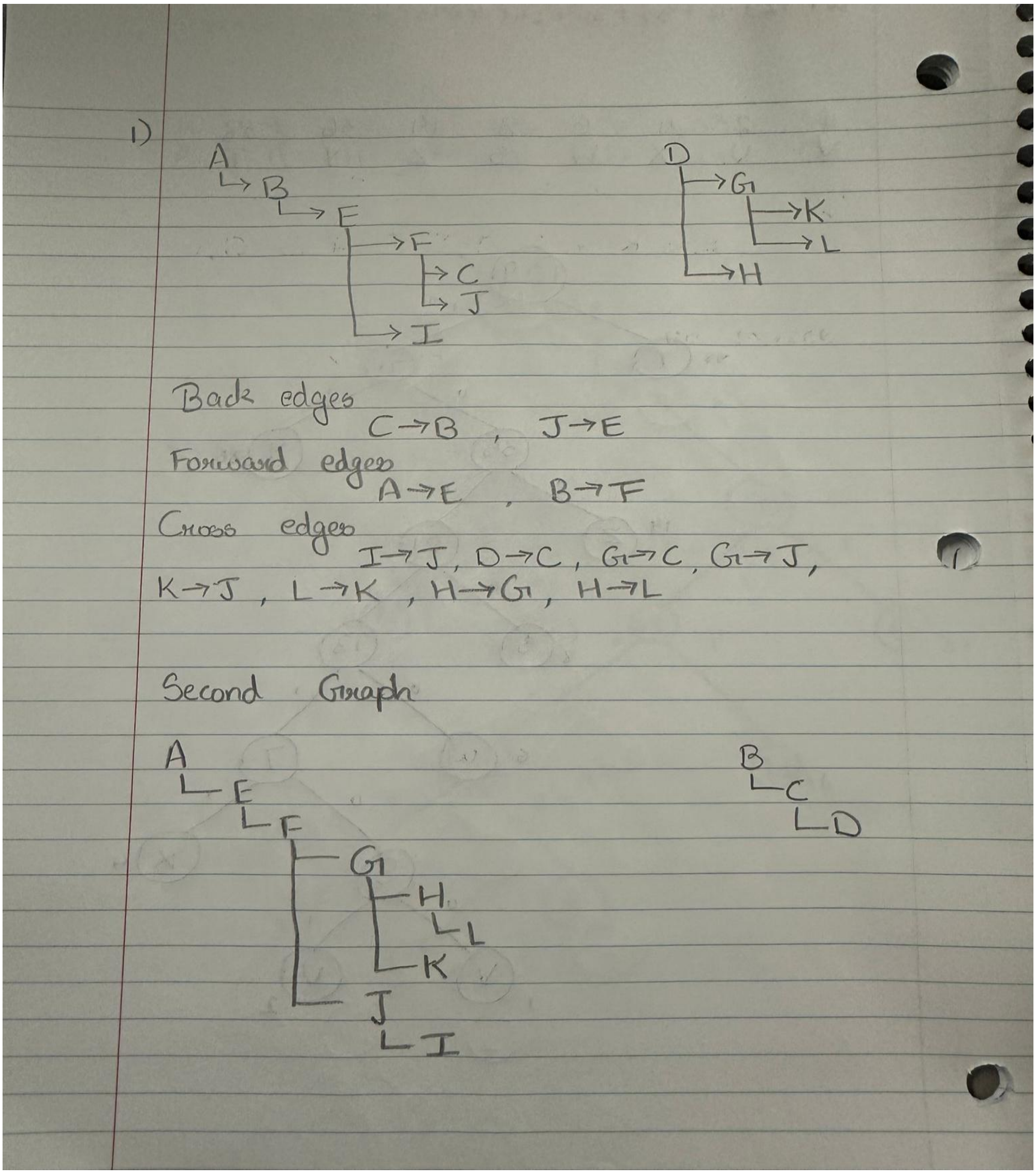
Name: Grishma Maheshbhai Thumar

Student ID: 922950012

Course: CSC 510, Spring 2025

Project: Homework 6

1. **DFS tree**



1)

- A
  - B
    - E
      - F
        - C
        - J
      - I

- D
  - G
    - K
    - L
  - H

Back edges
C → B , J → E

Forward edges
A → E , B → F

Cross edges
I → J , D → C , G → C , G → J ,
K → J , L → K , H → G , H → L

Second Graph

- A
  - E
    - F
      - G
        - H
          - L
        - K
      - J
        - I

- B
  - C
    - D

Pre-order

A, E, F, G, H, L, K, J, I, B, C, D

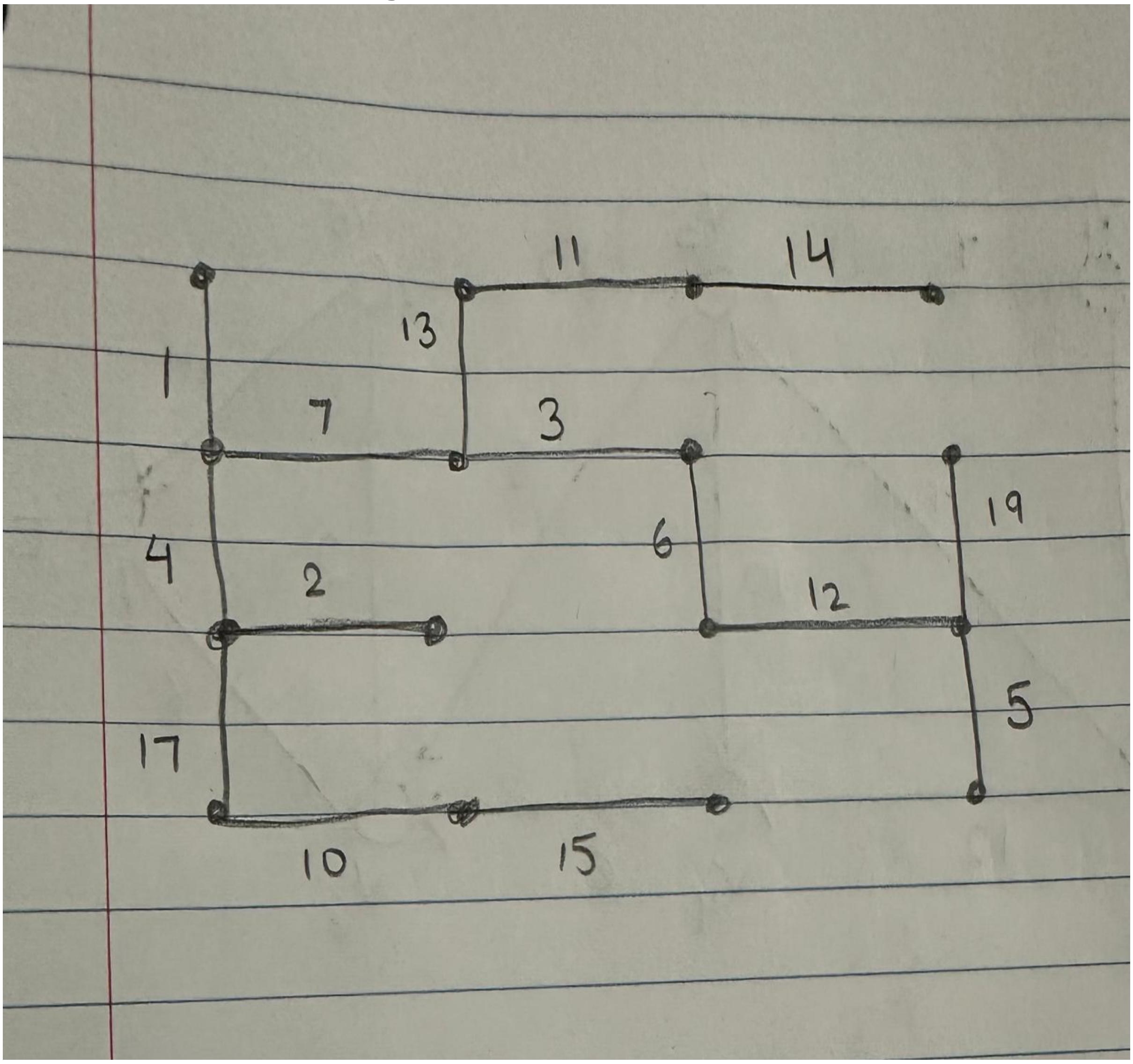
Post-order

L, H, K, G, I, J, F, E, A, D, C, B

Topological sort

B, C, D, A, E, F, J, I, G, K, H, L

2. **Kruskal and Prim's algorithms**



2) Kruskal's algorithm

- Sort all the edge weights
- Repeadetly pick the smallest edge without creating a cycle

So, the edges added in order are

1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 17, 19

It has 16 vertices, so we stop after 15 edges

Total weight = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 10 + 11 + 12 + 13 + 14 + 15 + 17 + 19
= 139

Prim's algorithm

- Initialize T = {C(1,1)}
- Repeatedly grow T by adding smallest edge leaving T

Chosen edges are:
10, 15, 17, 2, 4, 1, 7, 3, 6, 12, 5, 13, 11, 14, 19

Total weight = 10 + 15 + 17 + 2 + 4 + 1 + 7 + 3 + 6 + 12 + 5 + 13 + 11 + 14 + 19
= 139

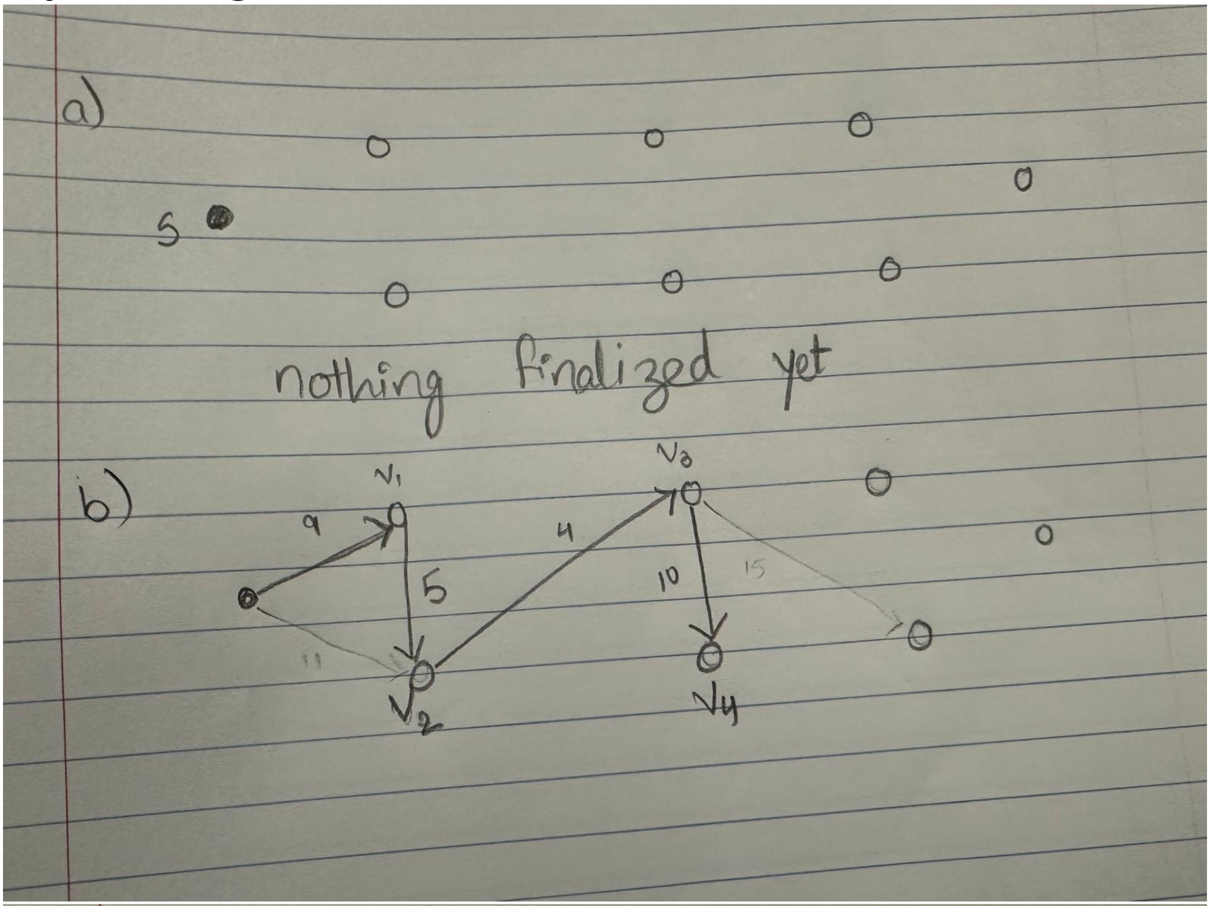
3.

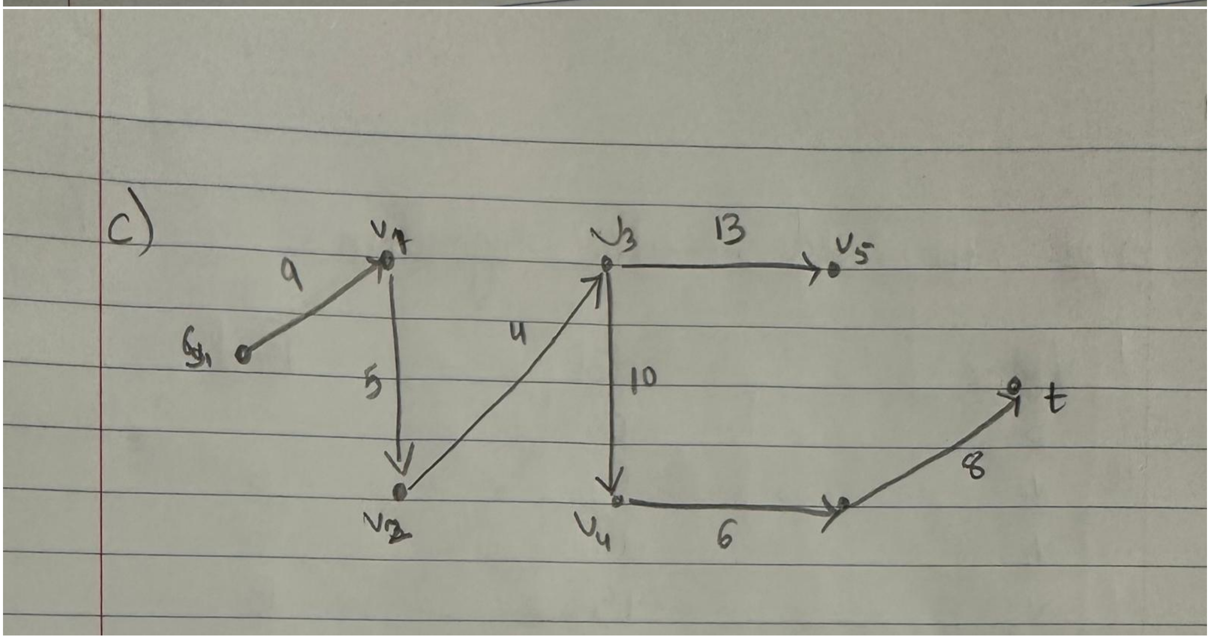
a) Yes, every MST must contain the lightest edge**.** Let e∗be the globally lightest edge in the graph, and consider the cut (S,V\S) that separates its two endpoints. By the **Cut Property**, the minimum-weight edge crossing *any* cut is "safe" to add to the MST. Since e∗e^*e∗ is the cheapest edge crossing this particular cut, it **must** appear in every MST.

b) Yes, if and only if that heaviest edge is a **bridge** (i.e.\ the *only* edge crossing some cut). Equivalently, by the **Cycle Property**, the maximum-weight edge on a cycle is never in the MST—but if the heaviest edge isn't on any cycle (so removing it disconnects the graph), it cannot be excluded without losing connectivity.

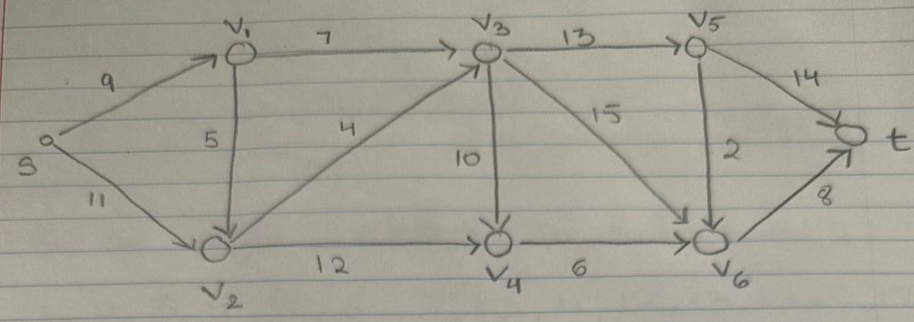
Example: Take two densely connected subgraphs (so every vertex still has degree ≥ 2) and join them by a single, very heavy edge. That edge is the unique crossing of the cut between the two parts, so the Cut Property forces it into the MST—even though its weight is the global maximum.

4. **Dijkstra's algorithm**





4)



a) At the beginning of the algorithm

dist= distance from s
pred= parent in SSP tree

Vertex	dist	Pred
s	0	-
$v_1, \ldots v_6$	∞	-
t	∞	-

b) After processing four vertices

Vertex	dist	Pred
s	0	-
v_1	9	s
v_2	11	s
v_3	15	v_2
v_4	23	v_2
v_5	28	v_3
v_6	30	v_3
t	∞	-

c) At the end of the algorithm

Vertex	dist	Pred
S	0	-
V_1	9	S
V_2	11	S
V_3	15	B
V_4	23	B
V_5	28	C
V_6	29	D
t	37	F