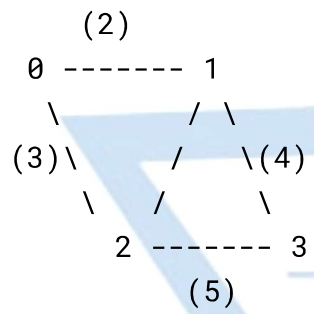
Prim's Algorithm Examples

Example 1:

Given Weighted Undirected Graph:



Edge List:

```
(0, 1, 2)
(0, 2, 3)
(1, 2, 1)
(1, 3, 4)
(2, 3, 5)
```

Number of vertices:

V = 4 (vertices numbered 0 to 3)

Use Prim's Algorithm to find the MST and calculate its minimum total weight.

Step-by-Step Execution (Starting from Node 0):

Initialization:

- Start node: `0`
- `Visited = {false, false, false, false}`
- `MST Weight = 0`
- Priority Queue (Min Heap) = `[(0, 0)]`

Step 1: Pick node `0` with weight `0`

- Add to MST → `MST Weight = 0`
- Mark `0` as visited → `{true, false, false, false}`
- Push adjacent edges:
  - `(0, 1, 2)` → push `(2, 1)`
  - `(0, 2, 3)` → push `(3, 2)`
- Heap: `[(2, 1), (3, 2)`

Step 2: Pick node 1 with weight 2

- Add to MST → `MST Weight = 2`

- Mark 1 as visited → `{true, true, false, false}`
- Push adjacent edges:
  - `(1, 2, 1)` → push `(1, 2)`
  - `(1, 3, 4)` → push `(4, 3)`
- Heap: `[(1, 2), (3, 2), (4, 3)`

Step 3: Pick node 2 with weight 1

- Add to MST → `MST Weight = 2 + 1 = 3`
- Mark 2 as visited → `{true, true, true, false}`
- Push edge `(2, 3, 5)` → push `(5, 3)`
- Heap: `[(3, 2), (4, 3), (5, 3)`

Step 4: Pick node 3 with weight 4

- Add to MST → `MST Weight = 3 + 4 = 7`
- Mark 3 as visited → `{true, true, true, true}`
- All vertices visited

Final MST:

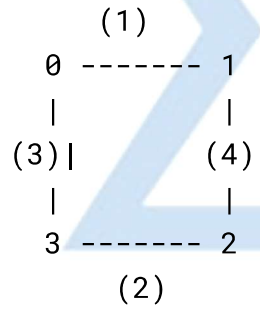
Included edges:

- (0 → 1) with weight 2
- (1 → 2) with weight 1
- (1 → 3) with weight 4

Total Minimum Spanning Tree Weight = 7

Example 2:

Graph Structure:



Edge List Representation:

```
(0, 1, 1)
(0, 3, 3)
(1, 2, 4)
(2, 3, 2)
```

Number of Vertices:

V = 4 (0 to 3)

Find MST using **Prim's Algorithm**, starting from vertex `0`.

Step-by-Step Execution:

Initialization:

- Start from vertex `0`
- `Visited = {false, false, false, false}`
- `MST Weight = 0`
- Priority Queue = `[(0, 0)]` → (weight, vertex)

Step 1: Pick `(0, 0)`

- Include vertex `0`, mark visited
  → `Visited = {true, false, false, false}`
- Add edges:
  - `(0, 1, 1)` → push `(1, 1)`
  - `(0, 3, 3)` → push `(3, 3)`
- MST Weight = 0
- Heap: `[(1, 1), (3, 3)]`
-

Step 2: Pick (1, 1)

- Include vertex 1, mark visited
  → `Visited = {true, true, false, false}`
- Add edge:
  - `(1, 2, 4)` → push `(4, 2)`
- MST Weight = 0 + 1 = 1
- Heap: `[(3, 3), (4, 2)]`

Step 3: Pick (3, 3)

- Include vertex 3, mark visited
  → `Visited = {true, true, false, true}`
- Add edge:
  - `(3, 2, 2)` → push `(2, 2)`
- MST Weight = 1 + 3 = 4
- Heap: `[(2, 2), (4, 2)]`

Step 4: Pick (2, 2)

- Include vertex 2, mark visited
  → `Visited = {true, true, true, true}`
- MST Weight = 4 + 2 = 6
- All vertices are now connected

Final Result:

MST Includes Edges:

- (0 – 1) → weight 1
- (0 – 3) → weight 3
- (3 – 2) → weight 2

Total MST Weight = 6