**Lab Report – Week 11**

CS2023 Data Structures and Algorithms

Dept. of Computer Science and Engineering, University of Moratuwa

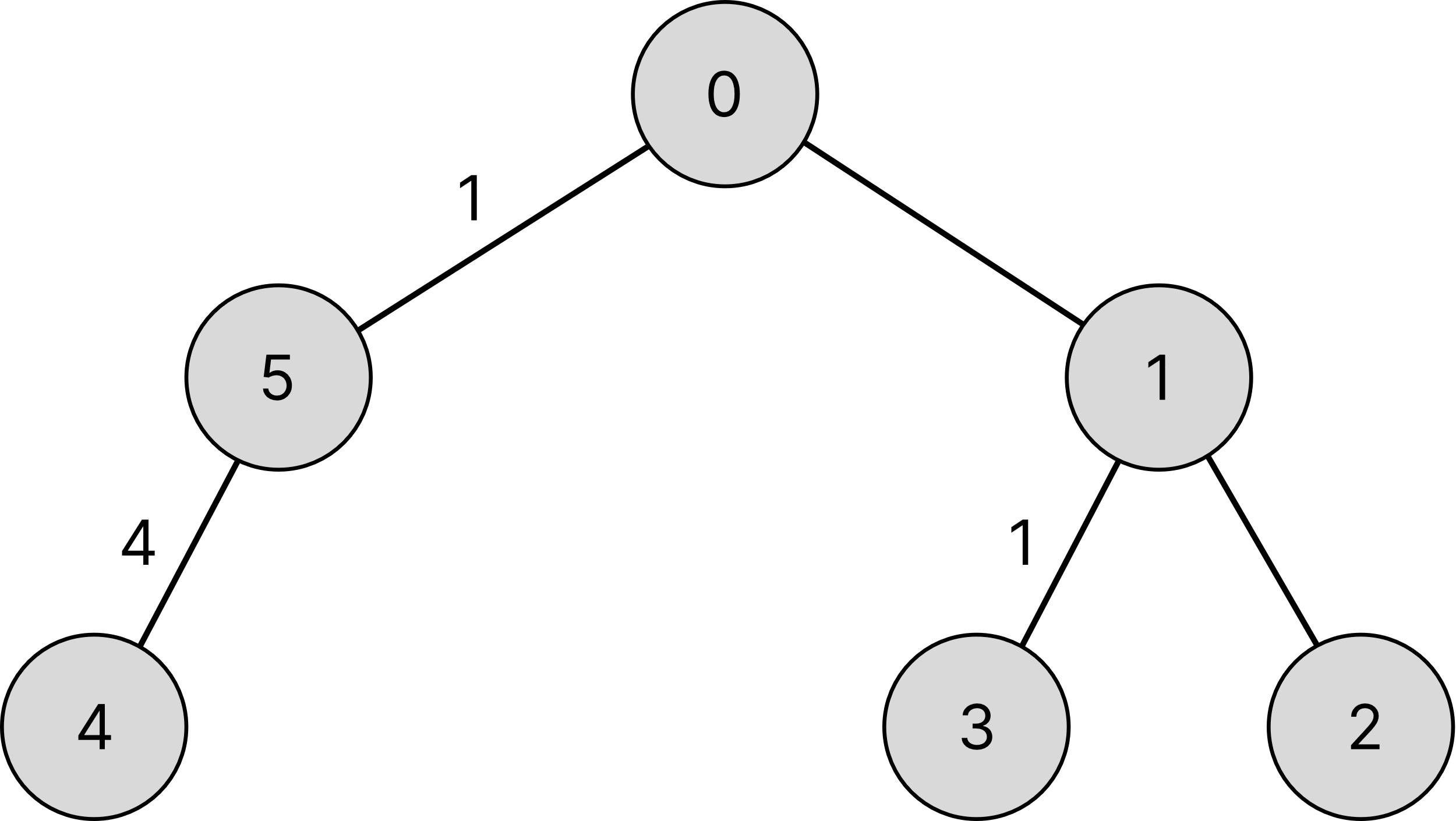
|  |  |
| --- | --- |
| Name: Tharindu Perera | Index Number: 210472J |

|  |
| --- |
|  |

### 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 0 | 3 | 0 | 0 | 0 | 1 |
| 1 | 3 | 0 | 2 | 1 | 10 | 0 |
| 2 | 0 | 2 | 0 | 3 | 0 | 5 |
| 3 | 0 | 1 | 3 | 0 | 5 | 0 |
| 4 | 0 | 10 | 0 | 5 | 0 | 4 |
| 5 | 1 | 0 | 5 | 0 | 4 | 0 |

2.



3.

A screen shot of a computer

Description automatically generated with medium confidence

4. Yes, MST in Question 2 and 3 are the same.

Edge weights have to be distinct.

5. The time complexity of Prim's algorythem, when using a priority queue to extract the minimal weighted edge, is either O(E log V) or O((V + E) log V). Here, V stands for the quantity of vertices, while E stands for the quantity of edges.

The sorting step in Kruskal's algorithm normally takes O(E log E) time, and the unionfind operations can be thought of as having an O(E log V) time complexity. So, either O(E log E) or O(E log V) can be used to express the overall time  complexity.

Prim's algorithm typically performs better than Kruskal's approach for dense graphs, whereas Kruskal's technique performs better for sparse graphs.

GitHub Link : [Tharindu6516/UoM-DSA-S2-Labs (github.com)](https://github.com/Tharindu6516/UoM-DSA-S2-Labs)