11210CS 431102 計算方法設計 Design and Analysis of Algorithms

Student: 葛洋晴 ID: 108006205

Bonus 4 Report

1. UNION-FIND Time complexity analysis:

To analyze the time complexity of the Union-Find class, we need to consider the two main operations it performs: ***find*** and ***union***.

Find:

Worst Case: O(N) for N elements.

Union: The union operation involves two find operations (to find the roots of the two sets) and then a constant-time operation to link the roots. Therefore, the time complexity of union largely depends on the time complexity of find. O(N).

2. KRUSKAL'S Algorithm with UNION-FIND analysis:

I. Sorting the Edges: O(ElogE).

II. Find and Union: O(N) for both

II. Iterating Over Edges: Kruskal's Algorithm iterates over all edges. For each edge, it performs at most two find operations and possibly one union operation. Therefore, the total time complexity for processing all edges is O(E·(N)).

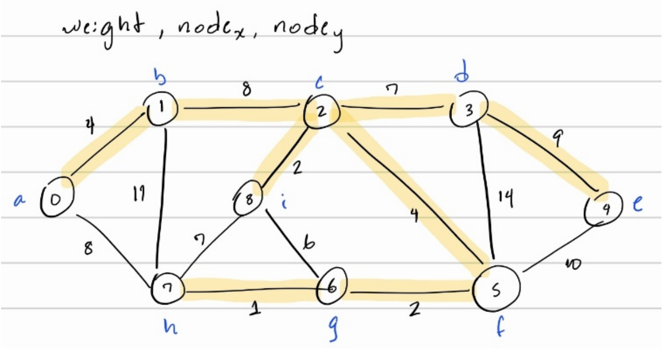
IV. Kruskal's Algorithm with Union-Find is: O(ElogE)+O(E·(N))

3. Edges selection step by step:

In this report I am going to put the order of the nodes, based on my algorithm. You can see more details such like the weight between the nodes and the MST's weight sum in the printed solution of 108006205_bonus.py script.

Edge Selection:

Step 1: (g, h),
Step 2: (f, g),
Step 3: (i, c),
Step 4: (a, b),
Step 5: (c, f),
Step 6: (c, d),
Step 7: (b, c),
Step 8: (d, e).


weight , $node_x$, $node_y$
b
c
d
8
7
4
1
2
3
9
2
a
0
11
8
i
14
4
9
e
8
7
6
10
7
6
5
1
2
h
g
f