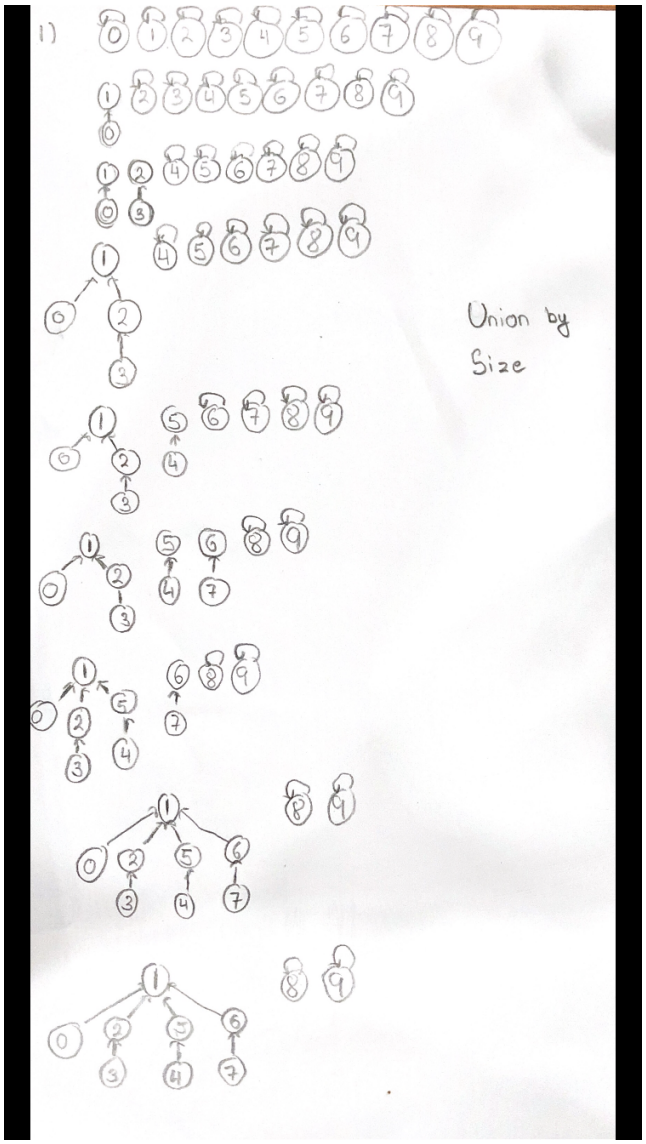
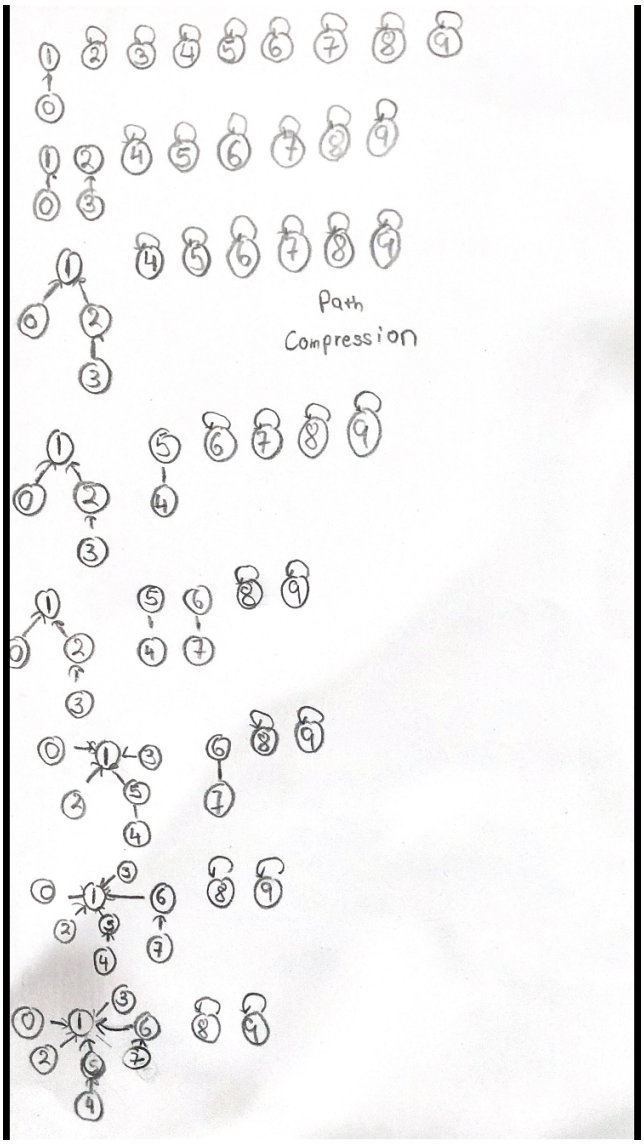
ICCS200: Assignment 7

Hasdin Ghogar
Collaborators: 1408
The date

1: Union By Hand





2: Breadth-First Search Running Time

```java
import java.util.Arrays;
import java.util.Set;
import java.util.TreeSet;

public class exercise2 {

    Set<Integer> nbrsExcluding(UndirectedGraph G, Set<Integer> vtxes, Set<In
        Set<Integer> union = new TreeSet<>();
        for (Integer src : vtxes) {
            for (Integer dst : G.adj(src))
                if (!excl.contains(dst)) {
                    union.add(dst);
                }
        }
        return union; }


    Set<Integer> bfs(UndirectedGraph G, int s) {
        Set<Integer> frontier = new TreeSet<>(Arrays.asList(s));
        Set<Integer> visited = new TreeSet<>(Arrays.asList(s));
        while (!frontier.isEmpty()) {
            frontier = nbrsExcluding(G, frontier, visited);
            visited.addAll(frontier);
        }
        return visited; }

}
```

nbrsExcluding loop takes O(m+nlogn) as the cost of it is to walk using bfs and the log n comes from the add operation. Since the most expensive call comes from frontier in bfs which takes running time not more than nbrsExcluding and visited.addAll takes only O(nlogn) therefore the code takes O(m+nlogn).