Data Structures and Algorithms

Agenda

- Time/Space complexity
- Union Find Algorithm
- Greedy Approach
- Kruskal's Min Spanning Tree
- Prim's Min Spanning Tree

Time/Space complexity

- Space complexity of Adjacency Matrix implementation: O(V*V)
- Space complexity of Adjacency List implementation: O(V+E)
- Time complexity of DFS/BFS
  - Adjacency Matrix implementation: O(V*V)
  - Adjacency List implementation: O(V+E)

    ```
    boolean[] marked = new boolean[];
    Stack<Integer> s = new Stack<>();
    s.push(start);
    marked[start] = true;
    while(!s.isEmpty()) {
        trav = s.pop();
        System.out.println(trav + ", ");
        for(Integer v:adjlist[trav]) {
            if(!marked[v]) {
                s.push(v);
                marked[v] = true;
            }
        } // max total itrs (inner loop) = 2E (for undirected) or E (for directed)
    } // max itrs (outer loop) = V
    ```

- Aux Space of DFS/BFS:
  - Max size of Stack/Queue = O(V)