Homework 7

1. Simulate Kruskal's algorithm implemented using disjoint sets to find a minimum spanning tree of the graphs given in parts *a* and *b* of Exercise 9.2.1 in the textbook (pages 331-332).

A1: Work done in this link

https://drive.google.com/file/d/1TP3Ix85MqzXeRPOYKG3CuEzhyYKlgLDa/view?usp=sharing

2. Simulate Dijkstra's algorithm implemented using min heaps to find shortest paths from source vertex *a* in the graphs given in parts *a* and *b* of Exercise 9.3.2 in the textbook (page 337). Treat each undirected edge as two directed edges.

A2: Work done in this link

https://drive.google.com/file/d/12Rd-fw9w9EqXs73IXeYftsMzsobs3yxC/view?usp=sharing

3. Exercise 9.2.5 in the textbook (page 332).

5. Design an algorithm for finding a maximum spanning tree—a spanning tree with the largest possible edge weight—of a weighted connected graph.
A3: Adapted from the notes, do the opposite of Kruskal's algorithm for minimum spanning trees; Choose the max weight at every step

```
wgraph kruskal(wgraph G)
{
     ds<int> D;
     For (auto v: G.V)
     ds.make_set(v);
     sort_Graph(G.E, DECREASING);
     // some sorting function in DECREASING mode
     // decreasing order of weight
     wgraph ans;

     for (auto e: G.E)
          if (ds.join(e.first, e.second))
               ans.add_edge(e);
     return ans;
}
```