**CS 6301.002. Implementation of advanced data structures and algorithms**

**Spring 2016**

Long project 2

**Group 10**

Vijay Mungara

Malav Shah

Madhuri Abnave

Gaurav Ketkar

**Abstract**: Write algorithms for finding minimum spanning trees. You must complete either level one (Prim1, Prim2, Kruskal), or level two (Edmonds' branching algorithm [MST in directed graphs]).

**Problem:**

Implement the algorithm discussed in class for finding minimum spanning trees in directed graphs. The input format is the same as level 1 (use readGraph method to create the graph). Assume that the root is vertex 1.

The algorithm for implementation is the same as explained in class

1. Transform weights so that every node except r has an incoming edge of weight 0:

2. Let G0 = (V, Z) be the subgraph of G containing all edges of 0-weight. Run DFS/BFS in G0, from r. Note that we are using only edges of G with 0-weight. If all nodes of V are reached from r, then return this DFS/BFS tree as MST.

3. If there is no spanning tree rooted at r in G0, then there is a 0-weight cycle. Find a 0-weight cycle

4. Shrink cycle C into a single node c. There may be many edges from the nodes of C to a node u outside the cycle. These are replaced by a single edge. For each edge (a, u) in G, with a ∈ C and u 6∈ C, introduce the edge (c, u) of weight w(a, u). Similarly, for edges of G that are going into C, do the following.

5. Recursively find an MST of the smaller graph. This MST has exactly one edge into c, and this edge corresponds to some actual edge (u, a) in the graph before shrinking, where a ∈ C. Now, expand node c, and include the edges of the 0-weight cycle C. Since the total weight of the cycle is 0, adding it to the MST does not increase its weight. But node a will have 2 incoming edges: edge (u, a) from the MST, and one edge from the cycle. Delete the edge coming into node a in the cycle, to get an MST of the original graph.

Return this MST.

**Output observations:**

**File Answer time**

1-lp2.txt 180922 2ms

2-lp2.txt 178316 8ms

3-lp2.txt 244483 17ms

lp2-ck 4964126 38140 ms