

**THE UNIVERSITY OF TEXAS AT DALLAS**



**LONG PROJECT 2**

**JAYAKARTHIGAYAN SRIDHARAN – jxs143730**

**SATHYANARAYANAN SRINIVASAN – sxs142031**

**SAAGARIKHA SRINIVASAN – sxs153930**

**CS 6301.002 Implementation of advanced data structures & algorithms**

**G19**

**Aim:**

To implement Edmond’s Branching algorithm in order to find Minimum Spanning Tree in a Directed Graph.

**Algorithm Flow:**

1. Transform incoming edges’ weights of each vertex relative to the weight of the minimum incoming edge to that respective vertex. Thereby producing zero weight incoming edge for each vertex.
2. Fetch all the vertices that are not reachable from the root vertex through the zero edges.
3. If all the vertices are reachable from root vertex through zero edges, then print the required output. Else find the zero edge cycles in which the non-reachable vertices are part of.
4. Shrink the zero edge cycle in to a super vertex.
5. Repeat from step 2 to 4 until all the vertices are reachable from the root vertex though the zero edges.
6. If the number of vertices is less than or equal 50, then expand all the super vertices starting from the latest one to the oldest super vertex and then print the MST along with the zero edges in the order of the TO vertices. Else print only the MST value.

**Instructions:**

**To Compile:** javac EdmondBranching.java

**To Execute:** java EdmondBranching [input file name]

**Results:**

Executed the code against various inputs provided for the LP2 level 2 and obtained the following values for MST and running times.

|  |  |  |
| --- | --- | --- |
| Input File Name | MST Value | Running Time (millisecond) |
| 0-lp2.txt | 17 | 2 |
| 1-lp2.txt | 180922 | 9 |
| 2-lp2.txt | 178316 | 15 |
| 3-lp2.txt | 244483 | 34 |
| lp2-ck.txt | 4964126 | 50693 |
| lp2-m.txt | 50324411 | 249979 |
| lp2-t1.txt | 46388 | 23 |
| lp2-t2.txt | 473197 | 1577 |
| lp2-t3.txt | 4894010 | 9977 |
| lp2-t4.txt | 50450242 | 67507 |

**Conclusion:**

Implemented Edmond’s Branching algorithm code is able to output the Minimum spanning Tree and its respective weight in a Directed graph.