



Design & Analysis of Algorithms

CS 302

Semester Project

Submitted to:

Sir Ahmed Nawaz

Submitted by:

Zubaria Ayub (17i-0231)

Muhammad Haim (17i-0108)

Section:

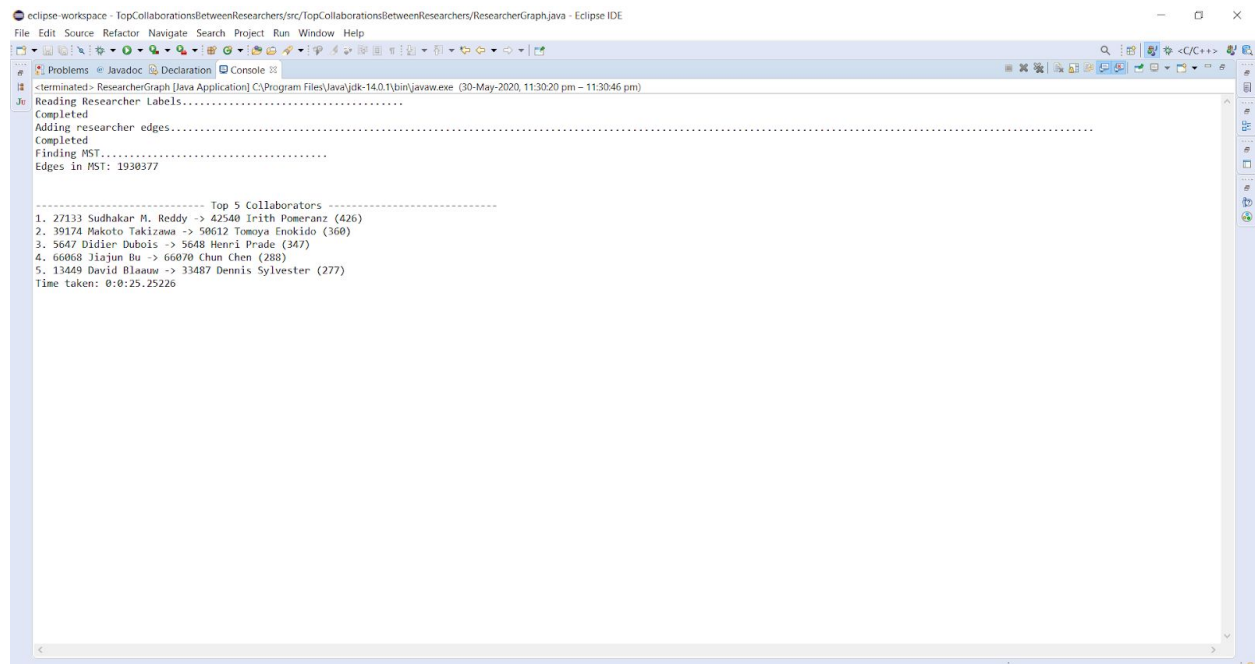
CS06-A

Department of Computer Science

National University of Computers & Emerging Sciences, Islamabad

Output	2
Algorithms	2
Graph Creation	2
MST Creation	2
Algorithm	2
Dry Run	3
Get Top Collaborators	3

Output



```
eclipse-workspace - TopCollaborationsBetweenResearchers/src/TopCollaborationsBetweenResearchers/ResearcherGraph.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
-terminated- ResearcherGraph [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (30-May-2020, 11:30:20 pm - 11:30:46 pm)
J:
  Problems Javadoc Declaration Console
  Reading Researcher Labels.....
  Completed
  Adding researcher edges.....
  Completed
  Finding MST.....
  Edges in MST: 1930377

----- Top 5 Collaborators -----
1. 27133 Sudhakar M. Reddy -> 42540 Irith Pomeranz (426)
2. 39174 Makoto Takizawa -> 50612 Tomoya Enokido (360)
3. 5647 Didier Dubois -> 5648 Henri Prade (347)
4. 66868 Jiajun Bu -> 66870 Chun Chen (288)
5. 13449 David Blaauw -> 33487 Dennis Sylvester (277)
Time taken: 0:0:25.25226
```

Algorithms

Graph Creation

1. Read "assets/coauth-DBLP-proj-graph/coauth-DBLP-node-labels.txt" to get ids and names of all researchers involved i.e. the vertices of the graph. Two vertices cannot have the same id and name.
2. Read "assets/coauth-DBLP-proj-graph/coauth-DBLP-proj-graph.txt" to get all the edges where vertices are specified by id. Both source and dest vertex must exist in the graph. If they don't, they are added.

MST Creation

Algorithm

1. Sort all edges of the graph in descending order.
2. Create an integer array setNumber of size equal to no of vertices - *this represents the id (index of smallest vertex in the given set) of the set a vertex belongs to. If two vertices belong to the same set (they are connected), they will have the same value in setNumber*
3. For index 0 -> size of setNumber:

- a. `setNumber[index]=index`
4. For each edge `e`
 - a. If `setNumber[src]==setNumber[dest]` - *adding e to graph will create a cycle*
 - i. ignore the edge
 - b. Else
 - i. add edge to graph
 - ii. `smallerSetId=min(setNumber[src],setNumber[dest])`
 - iii. `setNumber[src]=setNumber[dest]=smallerSetId`

The resulting graph is a maximum spanning tree

Dry Run

Initial Values

Index	0	1	2	3	4	5	6	7	8	9
Value	0	1	2	3	4	5	6	7	8	9

Add Edge between 0 and 9 - update value at index 9

Index	0	1	2	3	4	5	6	7	8	9
Value	0	1	2	3	4	5	6	7	8	0

Add Edge between 0 and 8 - update value at index 8

Index	0	1	2	3	4	5	6	7	8	9
Value	0	1	2	3	4	5	6	7	0	0

Add Edge between 8 and 9 - edge not added as `array[8]==array[9]` already ie they are already connected

Index	0	1	2	3	4	5	6	7	8	9
Value	0	1	2	3	4	5	6	7	0	0

Get Top Collaborators

1. Get an array of all edges in MST
2. Sort edges in descending order
3. Print first n edges from the array