

# Profiler Report

## Phase 1:

- Task: Generate the social network graph, store it in a graphml file.
- Times taken:

S.no	function name	Description	s/call	%of total time	no of calls
1.	main	the main function which calls the timekeeper and the generator	-	100	-
2.	tkmain(_fini)	main of thetimekeeper	230.21ns	100	477866
3.	initGenerateCourse	Generates the courses	-	90.92	-

1. In the generator, only generate courses is being observed, as it is the main thread.

## Phase 2:

- Task: With the .graphml file as input, answer the queries.
- The following was observed with 1700 students and 210 faculty.
- Times taken:

S.no	function name	Description	s/call	%of total time	no. of calls
1.	flWr()	Implements the floyd warshall algorithm	34.69	98.44	1
2.	Spath()	Moves along the shortest path for students and increments importance by 1	0.00	1.38	2890000
3.	impCal()	Calculates the importance of each person	0.53	0.09	1
4.	FPath()	Moves along the shortest path for faculty and increments importance by 1	0	0.03	44100
5.	parseGraphml()	loads the graphml file	0.03	0.00	1

1. Here Floyd warshall is taking the most of the time, which can be explained because it is  $O(|V|^3)$ , and here the size of the students is 1700 students,

which explains the large amount of time taken by the function. When, it was ran with a lesser number of students (450 to be precise), it took around 96% of the total time.

2. In the Spath and FPath, it iterates over the list of all shortest paths between the students and increases the importance of all the people on the path by 1(because they are on the shortest path), And so, the number of calls is the square of the number of students and faculty respectively.

## Team:

- V.A.V.S.Nikhil - 2012CS10260
- Nishant Kumar - 2012CS10239
- Mohit Agarwal - 2012CS10234