Project progress report

Title of project	**:** Fibonacci Heap
Team members	**:** Rushitkumar M Jasani (2018201034)
	: Priyendu Mori (2018201103)

Deliverables

Working fibonacci heap code with insert, getMinimum, deleteMinimum, decreaseKey. Also application of fibonacci heap in algorithms like prim's algorithm for minimum spanning trees and dijkstra's algorithm for single source shortest path will be delivered. And comparison of running time of above written algorithms with using fibonacci heap and using binary heap will be done.

Project delivery plan

Phase 1: Understanding the data structure and the scope of the project

Phase 2: Implementing and testing operations - insert, getMinimum, deleteMinimum, decreaseKey.

Phase 3: Implementing prim's algorithm for minimum spanning trees and dijkstra's algorithm for single source shortest path using fibonacci heap.

Phase 4: Testing the working of above implementation and comparing it's performance against binary heap implementation of same algorithms. Report and end user documentation writing.

Technologies to be used **:** c++

Online resources

https://en.wikipedia.org/wiki/Fibonacci_heap

https://www.growingwiththeweb.com/data-structures/fibonacci-heap/overview/

https://www.geeksforgeeks.org/fibonacci-heap-set-1-introduction/

Repository where work is being committed

https://github.com/rushitjasani/APS-Project

Plan for testing

1) Testing Fibonacci heap implementation

Writing scripts to generate test data (10 numbers, 50 numbers, 100 numbers...). Running them on both the fibonacci heap and binary heap and matching the output of all operations and comparing the running time for both.

2) Testing Algorithms that uses fibonacci heaps

Writing scripts to generate test data (10 nodes, 50 nodes, 100 nodes…). Running them on both the fibonacci heap and binary heap implementations of corresponding algorithms and comparing the running time for both.

End user documentation

Details about how to run the programs and how to perform various operations and check their outputs.