APS PROJECT REPORT

1. **Title of Project:**
   Fibonacci Heap

2. **Team Members:**
   Divyanshi Kushwaha (2018201046)
   Shreya Upadhyay (2018201091)

3. **Deliverables:**
   The idea is to implement successfully a Fibonacci Heap that enables certain operations such as Merge and ExtractMin in constant time along with other operations such as insert, delete and search for a KEY. Since Fibonacci Heap is a lazy and rather complicated implementation of Binomial heaps, it gives us major edge when the most frequent operations are the high speed operations which binomial heap does not do so efficiently. Thus our major deliverable is to implement these operations and also study the implementation of Fibonacci heaps in Dijkstra/ Prims/ Kruskal.

4. **Project Delivery plan:**

Functions	**Expected Completion Date**
INSERT operation in fibonacci heap	26/10/2018
FIND_MINIMUM in fibonacci heap	27/10/2018
DECREASE_KEY operation in fibonacci heap	29/10/2018
EXTRACT_MIN in fibonacci heap	1/11/2018
DECREASE_KEY in fibonacci heap	3/11/2018
**delete using decrease key in fibonacci heap	5/11/2018
Understand the application of fibonacci heap in existing standard algorithms	9/11/2018

**May be incorporated in a different function

5. **Technologies to be used:**
   - C/C++

6. **Online resources:**

- https://en.wikipedia.org/wiki/Fibonacci_heap
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwjArbbW16HeAhXI2LwKHS3qCd8QFjABegQICBAC&url=https%3A%2F%2Fwww.cs.princeton.edu%2F~wayne%2Fteaching%2Ffibonacci-heap.pdf&usg=AOvVaw3PZDCqclVlTdfxE58x2_Yv
- https://brilliant.org/wiki/fibonacci-heap/

7. **Repository where work is being commited:**
github: https://github.com/Divyanshi5/APS_Project

8. **Plan for testing and End User Documentation:**
Testing using rigorous test cases for all the functions.
Comparison with standard implementation and expected time complexity for each operation
Analysis for the algorithms.
sEnd User Documentation would contain how to compile the code and study of the applications.