Implementation of the van Emde Boas tree with application to Dijkstra and compare wrt Fibonacci and Binomial

Aishwary Dewangan
2018202016

Vishal Chugh
2018202021

- **Deliverables:** Implement Dijkstra using van Emde Boas Tree and compare this implementation with Dijkstra's implementation using Fibonacci Heap as well as using Binomial Heap.

- **Project Delivery Plan:**

Task	Expected Date of Completion
Implement Single Source Shortest Path Algorithm (Dijkstra's)	21st October 2018
Learn van Emde Boas Tree	23rd October 2018
Implement van Emde Boas Tree	25th October 2018
Implement Dijkstra's using van Emde Boas Tree	27th October 2018
Learn Fibonacci Heap	29th October 2018
Implement Fibonacci Heap	31st October 2018
Implement Dijkstra's using Fibonacci Heap	2nd November 2018
Learn Binomial Heap	4th November 2018
Implement Binomial Heap	6th November 2018
Implement Dijkstra's using Binomial Heap	8th November 2018
Comparison of Dijkstra's implementation using van Emde Boas Tree with the above two's implementation	10th November 2018

- **Technologies to be used:** C++ or Python

- **Online Resources:**
    - Wikipedia: https://en.wikipedia.org/wiki/Van_Emde_Boas_tree
    - Introduction to Algorithms (CLRS)
    - Youtube: https://www.youtube.com/watch?v=hmReJCupbN

- **Repository:** https://github.com/aishwr/APS-Project (Private Repository)

- **Plan for testing and End User Documentation:**
    - Run multiple test cases on Dijkstra's implementation using van Emde Boas Tree to check whether it gives result according to the standard implementation of Dijkstra's Single Source Shortest Path Algorithm.
    - Run multiple test cases on Dijkstra's implementation using Fibonacci Heap to check whether it gives result according to the standard implementation of Dijkstra's Single Source Shortest Path Algorithm.
    - Run multiple test cases on Dijkstra's implementation using Binomial Heap to check whether it gives result according to the standard implementation of Dijkstra's Single Source Shortest Path Algorithm.