CS 5V81.001: Implementation of data structures and algorithms

Project 2

Akshay Thakare (ast140230)

Description: Implement indexed priority queues and use it to implement Prim's Algorithm.

Background:

Prim's Algorithm: Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph.

Priority Queue: Priority queue is an abstract data type which is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority.

Indexed Priority Queue: Here we associate an index with each of the element for faster access of that element. As we are changing the priorities of the vertices by using decreaseKey function, we need to maintain the correct positions of the indices and thus we implement PQIndex interface to keep track of the indices by implementing the putIndex and getIndex methods.

Classes in the Project:

1. **Graph**
   Class defines the Graph structure.

2. **MST**
   Class for implementing the Prim's Algorithm

3. **PQIndex**
   Interface defining the putIndex() and getIndex() methods.

4. **PriorityQueueIndexed**
   This class contains methods: insert, add ,remove ,deleteMin ,decreaseKey , min, percolateUp, percolateDown, buildHeap

- add(T x) / insert(T x): inserts x in the priority queue.
- remove() / deleteMin(): removes and returns min element from the priority queue.
- decreaseKey(): restore heap order property after the priority of x has decreased after doing insert(x) operation.
- percolateUp() / percolateDown(): restore heap order property after the priority of x has been decreased after doing insert(x) operation/deleteMin() operation.
- min(): returns the min element

Underlying Data Structure:

Array: Here array[1] element is the minimum priority element and array[2*i] and array[(2*i)+1] are the left and right child of array[i] element.

Output Table:

Sr No	Input File	Output(Weight of MST)	Elapsed Time (ms)
1	prim1	84950	0
2	prim2	110419	0
3	prim3	153534	15
4	prim_ck	7384476	128