# ADVANCED DATA STRUCTURES

# PROJECT REPORT FOR IMPLEMENTATION OF FINDING "N" POPULAR HASHTAGS USING MAX FIBONACCI HEAPS

**Name : Raveerna Movva**

**UFID : 6711-6069**

**Email : mravee.1234@ufl.edu**

## FUNCTION PROTOTYPES:

### I) Fibonacci Heap:

- public boolean isEmpty(); // Returns true if heap is empty, false otherwise.

- public void increaseKey(Node x,int k); // Increases the datafield ( frequency ) value for a heap node.

- public void insert(Node node,int count); // Insert node into top level( root level ) list of heap.

- public Node removeMax(); // Extract Maximum Node( the one with the maximum frequency ) from Fibonacci Heap and call consolidate from here.

- public void consolidate(); // Pairwise Combine Fibonacci Heap- checks if there are 2 trees in the heap having the same degree or not. If it finds a match, then it makes the tree having root node of less datafield the child of root node having higher datafield.

- public void link(Node y1, Node x); // Make y1 a child of x.

- public int size(); // Returns number of nodes in the heap which is the size of the heap.

- public void cut(Node x, Node y); // Remove child x from parent y and insert child in top level list of heap.

- public void cascadingCut(Node y); // Do a cascade cut upwards towards the root until a node whose childcut field (isMark) is false is encountered.

### II) Node Structure:

- public Node(String hashtags, int datafield); // Constructor of class Node containing data-hashtag and datafield.

- public final String getHashtag(); // Returns string hashtag.

- public final int getData(); // Returns datafield which is the frequency of the hashtag.