ADVANCED DATA STRUCTURES

PROJECT REPORT FOR FINDING N MOST POPULAR HASHTAGS ON SOCIAL MEDIA USING FIBONACCI HEAPS

Name: Meghana Madineni
UFID: 9197-8425
Email: meghaname@ufl.edu

FUNCTION PROTOTYPES

I) Reading input and storing in Max-Fibonacci Heap and Hash Table:

- void **readInput** (String inputFile) //Reads each line of input file and processes accordingly
- void **process** (String queryString) //Generates Fibonacci heap and N most popular hashtags
- void **writeFile** (File outputFile, String outputLine) //Writes N most popular hashtags to output file

II) Node:

- public final int **getKey**() //Returns key value of a node
- public final String **getHashtag**() //Returns hashtag string of a node

III) Fibonacci Heap:

- public boolean **isHeapEmpty**() //Returns true if the heap is empty and false otherwise
- public void **insert**(Node newNode, int key) //Insert node into a heap
- public Node **removeMax**() //Extracts maximum node from Fibonacci heap and call pairwiseCombine() from here
- protected void **pairwiseCombine**() //Pairwise combine the Fibonacci heap

- public Node **increaseKey**(Node current, int newKey) //Increase value of the node given and returns it
- protected void **cut**(Node child, Node parent) //Remove child from parent and insert child in top level list of heap

- protected void **cascadeCut**(Node child) //Do a cascade cut upwards towards the root until a node whose childCut field is false is encoutered

PROGRAM STRUCTURE

- The path to input file name is given as an argument to the program (hashtagcounter.java).

- The input file name is passed as an input to readInput() function of ProcessInput.java.

- Each line that starts with # is split into hashtag and frequency and a node is added to the Fibonacci heap if the hashtag is encountered for the first time using insert(). Otherwise the frequency is incremented using increaseKey() and cut() and cascadeCut() functions are invoked if needed.

- When a query is encounterd, removeMax() followed by pairwiseCombine() are called for query number of times and the output is written to output_file.txt using writeFile() and the removed nodes are added back to the Fibonacci heap.

- This continues till all the lines in the input file are read and the program terminates when “STOP” is encountered.