ADVANCED DATA STRUCTURE

PROJECT REPORT

Topic: FINDING N MOST POPULAR HASHTAGS ON SOCIAL MEDIA USING FIBONACCI HEAPS

Submitted to: Dr. Sartaj K Sahni

Submitted by:

Name: MAYANK SHARMA
UF ID: 98831646
Email: MAYANKSHARMA@UFL.EDU

FUNCTION PROTOTYPES

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

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

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** (Node newNode, int key) //Insert node into a heap
- public Node **f_removeMax**() //Extracts maximum node from Fibonacci heap and call pairwiseCombine() from here
- protected void **pairwiseCombine**() //Pairwise combine the Fibonacci heap
- public Node **f_incr_Key**(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 HashTagCounter.java

- Each line starting with # is divided into hashtag and frequency and if hashtag is seen for the first time a node is added to the Fibonacci heap using insert(). Else increaseKey() is used to increment the frequency and cut() and cascadeCut() functions are invoked when needed.

- removeMax() accompanied with pairwiaseCombine() are invoked query number of times if a query is encountered and the output is written to a outputFile whose name is specified by the user using the writeFile() function.

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

COMMAND LINE ARGUMENTS

- java HashTagCounter [input_file_name]

  This will write the output in the console

- java HashTagCounter [input_file_name] [output_file_name]

  This will create a new file with name a [output_file_name] and write the output in that file

- if the command will contain more than two arguments it will give a message of "Invalid input".