ADVANCED DATA STRUCTURES – PROGRAMMING ASSIGNMENT
COP 5536 Spring 2020

Name: Archit Khatri
UFID: 3333 9695
Email: architkhatri@ufl.edu

Compiler Information: Java Programming Language Compiler (javac) is used.
There are 2 class files used in this programming assignment.
FibonacciHeap.java and hashtagcounter.java

FibonacciHeap.java

First of all, FibonacciHeap class contains another class known as FibonacciHeapNode. This class contains the structure of the node of the Fibonacci heap. The attributes of the node are **child, left, right, parent, childCut, degree, name** and **elementValue**.

FibonacciHeap class then further provides the required functionality that a Fibonacci Heap using hash table must possess. Data includes the Root Fibonacci node and the Hash map created for lookups**.** The methods that are used in this class are:
hashTagsWriteWithOP, printHashtags, insert, combinePairs, removeMaxWithOP, removeMax, increaseKey, mergeTrees and constructors.

Insert():

This is the most basic functionality, it will create a node and then insert that node with the hashtag name into the Fibonacci Heap. Takes two argument, name of the hashtag and the value of the count of that hashtag. If the hashtag already exists in the hashtable, then the node would be passed on to the increaseKey() method.

removeMax():

There are two functions that provide the Remove Max functionality. If the output file is provided then removeMaxWithOP will be called because it contains parameter for the output file name and if the output file isn't provided then the removeMax will be called which will then simply display the hashtags. Although both the functions perform the main task that is to remove the maximum recursively until mentioned explicitly. Everytime a max node is removed, it is stored in a different node. After removing max there is need to do a pairwise combine so, combinePairs method is called. Since the maximum node is removed, there is need to store or print the hashtag into a file or the console and therefore hashTagsWriteWithOP and printHashTags functions are called.

printHashTags():

This will simply display the max hashtags that were removed from the Fibonacci heap onto the user's terminal which is the desired output.

combinePairs():

combinePairs method will do a pairwise combine on the Fibonacci Heap after the max node is removed and then a new hashmap is created that will keep the track of new degree of the nodes. This function will continue until we reach the first node from where we started.

increaseKey():

This function will increase the value of the element if the current value of the element is greater than the parent node value. If the condition becomes true then the node is removed and becomes the root by inserting at that place. Then childCut values are checked and the childCut value is set to true even if it is cut cascade.

mergeTrees():

This function is used when the root is removed from the heap and the resulting child nodes are needed to be melded back together. Therefore it takes two arguments FibonacciHeapNode1 and FibonacciHeapNode2 and merges the tree together.

hashTagsWriteWithOP():

This function will write the max hashtags on the file that is provided by the user. The hashtags that are written on the file are the one which result from the root of the max Fibonacci heap.

hashtagcounter.java:

This is the main class which will execute the functions in the FibonacciHeap as per the desired requirements. This file will check for the input file and output file if provided and will display most popular hashtags on the console if the output file isn't provided else, it will create a file that was specified in the command line arguments and will write the most popular hashtags in that file.

It will use command line argument length to check if the output file is provided or not and based on this the desired piece of code from FibonacciHeap will execute.

The program can be run using two commands:

1. java hashtagcounter [input_file]

2. java hashtagcounter [input_file] [output_file]