

Advances Data Structures (COP 5536)
Programming Project Report

Name: Drona Banerjee
UFID: 46627749
UF Email: dr.banerjee@ufl.edu

PROJECT DESCRIPTION

The goal of this project is to implement a system to find the n most popular hashtags appeared on social media such as Facebook or Twitter. For the scope of this project hashtags are taken as an input file. Basic idea for the implementation is to use a max priority structure to find out the most popular hashtags.

The project uses the following data structure.

1. Max Fibonacci heap: used to keep track of the frequencies of hashtags.
2. Hash table : The key for the hash table is the hashtag, and the value is the pointer to the corresponding node in the Fibonacci heap.

PROGRAMMING ENVIRONMENT

The project has been implemented in Java. A makefile has been written which creates an executable file named hashtagcounter.

INSTRUCTIONS TO RUN

Compile the program using the makefile. Then run the following command:

```
java hashtagcounter <input_file_name> [output_file_name]
```

The output filename is optional. If an output filename is provided, a file is created with the name provided and the output is written to it. If the output filename is not provided, the output is printed in the console.

FUNCTION PROTOTYPES AND STRUCTURE OF THE PROGRAM

Function prototypes:

- **public void insertNode(FibonacciHeapNode<T> node, double nodeKey)**

This function inserts a node in the Max Fibonacci heap

- **protected void cutNode(FibonacciHeapNode<T> c, FibonacciHeapNode<T> p)**

Performs the cut operation to cut/remove child from parent

- **protected void cascadingCut(FibonacciHeapNode<T> n)**

Performs a cascading cut. We keep cutting if the node's parent has Child Cut value set as true

- **public void increaseKey(FibonacciHeapNode<T> n, double increaseVal)**

Increases the value of the key of a particular node in the fibonacci heap by the increase value.

- **public FibonacciHeapNode<T> removeMax()**

The max element is removed from the heap and degree-wise meld is performed

- **protected void degreeWiseMeld()**

Performs meld operation by doing a degree-wise merge

Program Structure:

1. The main method in the driver class hashtagcounter accepts command line argument with input file name and output file name(optional)
2. If output filename is not provided, the output is printed in the console. If output filename is provided, a file with the output filename provided is created and the output is written to it.
3. Each line is read until 'stop' is encountered.
4. If a valid Hashtag is encountered:

The program checks if it is already in the hashtable.

If yes – An increase key operation is performed

public void increaseKey(FibonacciHeapNode<T> n, double increaseVal)

Else – The node is inserted using insert node operation

public void insertNode(FibonacciHeapNode<T> node, double nodeKey)

5. If query integer (n) is encountered:

The n most popular Hashtags are printed/written by performing the remove max operation n times.

public FibonacciHeapNode<T> removeMax()

Each time the max node is removed, we perform a degree-wise meld using:

protected void degreeWiseMeld()

The removed nodes are then inserted back into the heap.

6. If 'stop' is encountered:

The program is terminated.