Project Report – HashTag Counter

Name: Rohit Karumuri

UFID: 90971158

Email: rohitkarumuri@ufl.edu

MakeFile instructions:

- make clean - for clean compiled files
- make – compiling files
- ./hashtagcounter <input> [output] – executing the program

Description and Structure:

The project is implemented using a maximum Fibonacci heap data structure on C++ language. The flow of the program is as explained below:

- The main receives the input file name.
- The App object processes each line of the input file and based on the given input, it either adds or increases count value of hashtags or removes the maximum hashtag values and prints them to either output file or console. Each name will be stored in a hash table (imported from Standard Template Library) in order to retrieve corresponding node fastly.
- The functions shown is step 2 are handled by the FibonacciHeap Class whose nodes are FHNode struct.

App Class

Attributes:-

- inFile – input file name
- outFile – output file name
- ifs - input file stream
- ofs - output file stream
- hashTable - hash table (imported from STL)
- fHeap - Fibonacci Heap object

Member Functions:-

- processLine - processes each line of input file.
- addToHashTable - adds the tag name and count to fibonacci heap and stores the tag names in the hashTable.
- printToOutput - prints maximum hashtag names to preferred output (console/outstream)

FHNode struct

- int count – count value of hash tag
- string tagName – name of hash tag
- FHNode *parent – pointer to parent
- FHNode *child – pointer to child
- FHNode *left – pointer to left
- FHNode *right – pointer to right
- int degree - number of direct children
- bool childCut – childcut value; true when child is lost, false when max removal

FibonacciHeap Class

Attributes:-

- maxNode - pointer to node of maximum value in heap
- numNodes - size of heap
- nodeMap – maintains a map of nodes for pairwise combination

Private Functions:-

- createNode - creates new node
- meldNodes – melds/ connects two nodes
- collectRoots – returns list(vector) of nodes of same degree
- addToNodeMap – adds element in nodeMap for pairwise combination
- pairwiseCombine – makes one node as parent of other node based on count value; also checks if child is null, it will meld with child node of other tree

Public Functions:-

- Insert – inserts new node in heap
- Meld – public extension of meld function
- GetMax – returns node of maximum count
- RemoveMax – returns node of maximum count and removes from heap; also does pairwise combine afterwards.
- SetMax – Sets maximum node
- IncreaseKey – increases count of exisiting node; becomes maxnode if it is maximum
- GetNumNodes – returns size of heap