Advanced data structure(COP5536)

Summer 2016

Project Report

Chengyun Xu

UFID# 4660-2243

cyxu.9416@ufl.edu

PROJECT DESCRIPTION

In this project, we need use two types of data structure to implement a system which can find the most popular hashtags in a text file. Max-Fibonacci heap and hashtable need to be used in this project. The Max-Fibonacci heap is used to record the frequency of a hashtag and the hashtable contains the hashtag and the node in Max-Fibonacci heap which store the frequency of hashtag, If we want, hashtable can be used directly from the library implementation, but Max-Fibonacci heap should implemented by our own primitive data structure. In the input file, each hashtag followed by its frequency begins with # sign. If there are the same hashtags, we need to increase the hashtag frequency. Query is also included in the input file in different lines without # in the beginning. We need to read a file as a format regarding the file name as an argument. The output should be separated by comma in a line. When a new query appears, we have to write down the hashtags which has the highest frequency. The output name should be “output_file.txt”.

FIBONACCI HEAP

A Fibonacci heap is a kind of heap which can be used to combine priority queue, it has better performance in amortized computation analysis. Comparing to binomial heap, insert, delete and union operation only cost O(1) amortized time. Otherwise, binomial heap need O(lgn) amortized time in delete and insert and O(n) amortized time in union operation. It has better performance than binomial heap.

STRUCTURE OF THE PROGRAM

I used three classes to implement the system

In the below part, I will give some descriptions to each method.

Node.java

the class defines the node property in FibonacciHeap

There are seven variables defined in this class

int key

Key stores the node value.

int degree

Degree stores the number of children that the node has

Node left

Left points to left sibling of the node.

Node right

Right points to right sibling of the node.

Node child

Child points to the child of the node.

Node parent

Patent points to the parent of the node.

boolean marked

The variable is used to define whether a child of the node is deleted.

String hashtag

The variable is used to store the hashtag in the node.

The class contains the function :

public Node(int key,String hashtag)

The function is used to initialize a node.

FibinacciHeap.java

The class constructs some common Max-Fibonacci heap methods. Some operations are needed to call the instance of class Node.

public FibonacciHeap()

This method is used to initialize a Fibonacci heap.

private void removeNode(Node node)

Return type: void

Parameters: Node node

This method is used remove the node and its children in a doubly linked list.

public void insertNewNode(Node node)

Return type: void

Parameters: Node node

This method is used to insert a new node is a Max-Fibonacci heap. If the the key of the node is larger than previous max node, then, update the max node to the new node and increase the number of the key

private void link(Node node, Node pnode)

Return type: void

Parameters: Node node,Node pnode

This method is used to add node as a child of pnode and increase the degree of pnode by one. It is necessary is some operations in Fibonacci heap.

private void pairwise()

Return type: void

Parameters: node

This method is used to do pairwise combine. We need to initialize a new arraylist that contains the nodes. First of all, in the Max-Fibonacci heap, if the degree of current node equals to the degree of node in the arraylist, we need to link them and add the max node of new tree in the arraylist. In the second step , we need to check whether the nodes of the array list have the same degree. If it does, we also need to combine them together. Finally, we need to find the right sibling of current node, and do the same operations as listed before.

public void removeMax()

Return type: void

Parameters: node

It is a relatively complicated method. The method is used to remove the max node in Max-Fibonacci heap and add the children of the max node to the root node doubly linked list. And then, we need to do pairwise combine until each node does not have the same degree with others in the

Max-Fibonacci heap. Finally ,we need to decrease the number of nodes by one.

public int getMaxKey()

Return type: max.key

Parameters: node

This method is used to get the key of the max node in the Fibonacci heap.

private void cut(Node x, Node y)

Return type: void

Parameters: Node x, Node y

This method is used to cut node x from y, and link x to the root. It is the first step of cut a node from Fibonacci heap.

private void cascadingCut(Node node)

Return type: void

Parameters: Node node

This method is used to do cascading cut in Fibonacci heap. If its parent did not lose child before(marked = false), we just need to cut the node mark the marked label to true, and all is done. If its parent lost child before(marked = true) , we also need to do cascading cut to its parent too.

public void increaseKey(Node p, int key)

Return type: void

Parameters: Node p, int key

This method is used to increase the key of the node of the Fibonacci heap.

If the new key of the node is bigger than before, we need to cut it from its parent and add it to the root doubly linked list. And we need to do cascading cut to its parent.

public Node getMaxNode()

Return type: Node

Parameters: none

This method is used to get the max node of the Fibonacci heap.

hashtagcounter.java

The class used to read the input file and call the method in FibonacciHeap to calculate the frequency of hashtag and get the most popular n hashtags.

public static void execute(String fileName)

Return type: void

Parameters: String fileName

In this method, the argument is delivered in to the method as a file name. We have some steps to do in this method.

1. Create hashtable and Max-Fibonacciheap.
2. Read a line of the input text.
3. If the input line equals “STOP” , we need to end the program.
4. If the first character equals “#” and the hashtag is not contained in the hashtable, we need to wrap the hashtag and frequency as a node and

insert it into Fibonacci heap.

5. If the first character equals “#” and the hashtag is contained in the hashtable, we need to update the frequency and do the increase key method to the node in Fibonacci heap.

6. If first character doesn’t equal “#”, we need to transfer the first character as a number. And then, we need to remove the max node in the Fibonacci heap n times (n equals the number) and get the hashtag and frequency.

7. We need to insert the node which we have removed before back.

8. Read the next line of the input and do the step1 again.

9. Output the file to output_file.txt.

The structure of execute are shown in figure 1.

public static void main(String[] args)

The main method is used to deliver the argument as file name to call execute(String fileName) method.

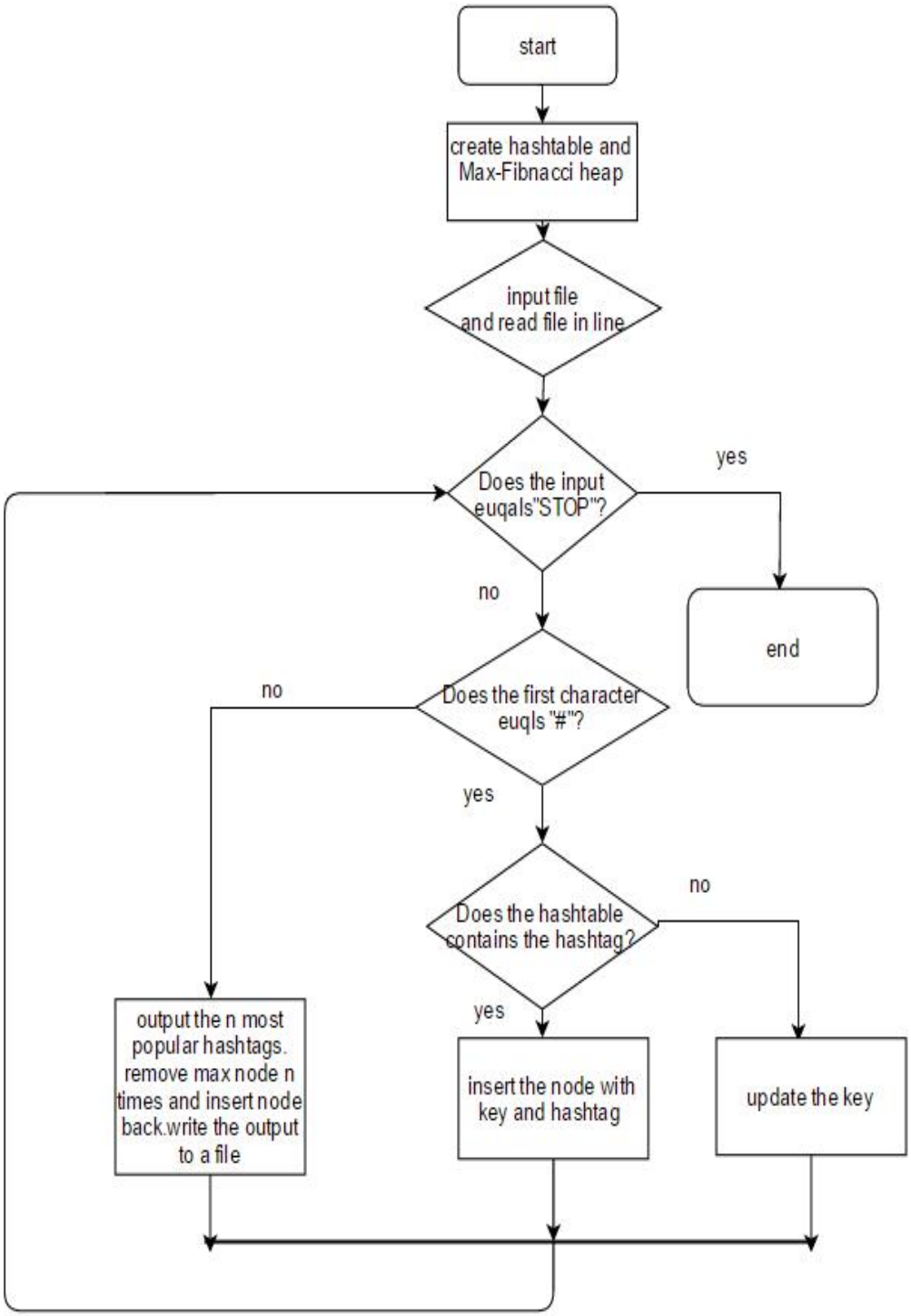

Figure 1. the structure of method execute()

THE RESULT OF PROGTAM

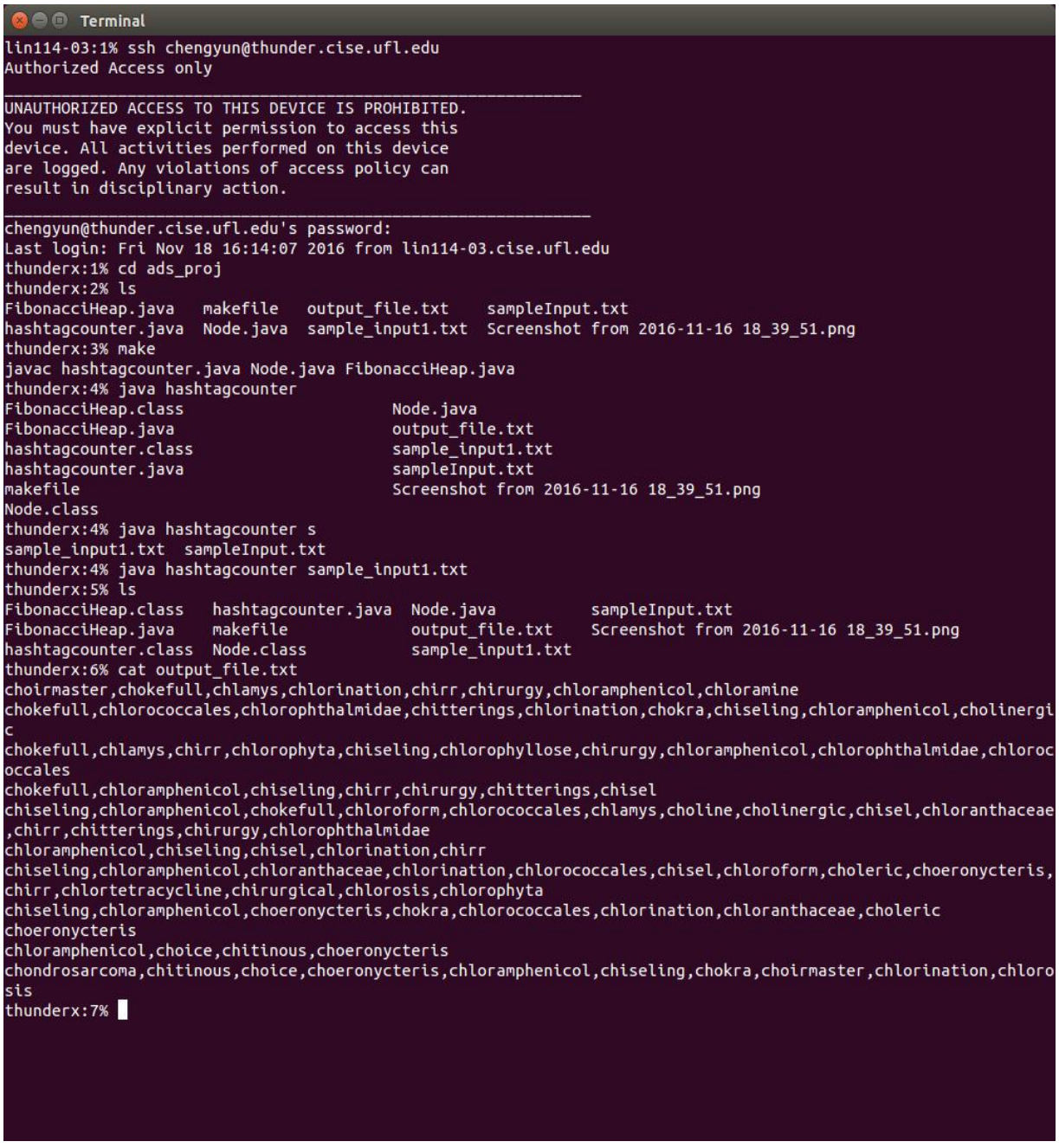

Figure 2. Output result

From the result, after we input the sample_input1.text, we can get the result what we want exactly.