Advanced Data Structures (COP 5536)

Fall 2018

Programming Project Report

Purva Kulkarni

UFID 82721493

purvakulkarni@ufl.edu

Problem Statement

A new search engine “DuckDuckGo” is implementing a system to count the most popular keywords used in their search engine. They want to know what the n most popular keywords are at any given time. You are required to undertake that implementation. Keywords will be given from an input file together with their frequencies. You need to use a max priority structure to find the most popular keywords.

Project Description:

This project aims at extracting the most popular keywords using max priority structure. The data structures used are Max Fibonacci Heap and Hash Maps. Max Fibonacci Heap is used to keep the track of the frequencies of the words. Words with the highest frequencies are extracted depending on the query which is to be performed.

Structure of the program and function prototypes:

Class Fib_heap have the following functions which are used in the Max Fibonacci heap implementation. Two hash maps and a vector are also used to store the data of the node ie. the string and the frequency.

Function	node* create_node(int data)
Description	This function is used to create a node.
Parameter	Int data : data of the newly created node
Return Value	node*

Function	node* RemoveMax()
Description	This function is used to remove maximum node from the heap structure.
Parameter	null
Return Value	node*

Function	node* meld(node *p, node *q)
Description	This function is used to meld two lists into one.
Parameter	node *p, node *q : pointer to nodes which are to be melded
Return Value	Node*

Function	void IncreaseKey(node* p, int data)
Description	This function increases the frequency value of the given node by key.
Parameter	Node*p, int data : the node whose data is to be increased.
Return Value	void

Function	node* insert(int data)
Description	This function is used to insert new node into the heap
Parameter	int data : The data to be in inserted in the node.
Return Value	Node*

Function	void printnode()
Description	This function is used to print the node
Parameter	null
Return Value	void

Function	int main(int c, char *v[])
Description	Read input file and write into output file
Parameter	int c, char *v[]
Return Value	0

Node Structure :

The node structure consists of data, degree, child, parent, left, right and childcut.
Class : Fib_heap

Variable	max_node : used to store maximum frequency
Datatype	Node*

Variable	p: used for traversing child nodes
Datatype	Node*

Variable	m: used for pairwise combine
Datatype	Node*

Variable	n : used for pairwise combine
Datatype	Node*

Variable	p in increase key : used to increase the value of node
Datatype	Node*

Variable	q : used for pairwise combine
Datatype	Node*

Conclusion : The project goal to implement Max Fibonacci heaps to find the most popular words has been successfully implemented.