COP 5536 : ADVANCED DATA STRUCTURES
PROJECT REPORT FILE

TOPIC: KEYWORD COUNTER

Submitted by -

Sugandha Gupta
UFID: 9768-9051
UF mail: sugandhagupta@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.
You must use the following data structures for the implementation.

- Max Fibonacci heap: to keep track of the frequencies of keywords
- Hash table: keywords should be used as keys for the hash table and value is the pointer to the corresponding node in the Fibonacci heap.

Function Prototypes:

class node

Description	Class defining the node structure of the Fibonacci Heap
Parameters	int frequency string keyword unsigned long int degree bool childcut node* left node* right node* child node* parent

void consolidate(node* c, node* p)

Description	To merge two trees of same degree (p>c)
Parameters	node* c child node* p parent
Return Value	No value returned

void insert(node* new_key, bool existing)

Description	Insert new node in heap
Parameters	new_key : to be inserted existing : check if it already exists
Return Value	No value returned

void cut(node* cutnode, node* cut_parent)

Description	Cut a node from the heap
Parameters	Cutnode : node that has to be cut Cut_parent : parent of the removed node
Return Value	No value returned

void meld_fib()

Description	Meld the heap after remove operation
Parameters	It keeps a table to track the nodes according to their degrees and sends it to consolidate function to join.
Return Value	No value returned

void cascade_cut(node* curr)

Description	Cuts out nodes whose childcut values become true
Parameters	curr node: Points to the parent of the removed node and keeps cutting nodes till it reaches a node where the value of childcut becomes false
Return Value	No value returned

void increase_frequency(node* changenode)

Description	Edits the frequency of a node
Parameters	Changenode : It's the node whose frequency is to be increased.
Return Value	No value returned

int main(int argc, char *argv[])

Description	Main function to run the code
Parameters	argc : Counts the number of arguments passed argv[] : Stores the arguments in an array Argv[1] in this case is the argument that accepts the name of the input file.
Return Value	Output_file.txt is the final file formed

Assumptions:

- Input keyword can be any arbitrary string including alphanumeric characters and special symbols.
- The count/frequency can only be a positive number greater than 0.
- No spaces in the keywords.
- For two keywords to be same, whole word should match. i.e. #youtube and #youtube_music are two different keywords.
- One query has only one integer.
- If there are more than one keyword with similar frequencies, you can print them in any order.
- Query integer *n*≤20. i.e. you need to find at most 20 popular keywords.
- Input file may consist of more than 1 million keywords.

Conclusion :

The project has been implemented successfully and tested on use cases.