COP5536 Fall 18 Programming Project

Neel Manish Rami
UFID : 7712-3151
UF Email : nrami@ufl.edu

November 16, 2018

1 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. Keywords are given from an input file together with their frequencies.

2 Project Structure

The project consists of 3 Java files:-

2.1 FHNode.java

This class defines the structure of a Max Fibonacci Heap Node.

2.2 MaxFHNode.java

This class implements a Max Fibonacci Heap and all the functions such as

- Insert Operation
- Increase Key Operation
- Remove Max Operation
- Cut Operation
- Cascading Cut Operation
- Pairwise Combine Operation

This class also maintains Hash Table where word is stored as key and value is the object reference of the node that is associated with the key.

2.3 DuckDuckGo.java

This class serves as the entry point of the project. Here input file name is taken as command line argument, then the input file is read and finally output file is generated.

3 Input and Output File Format

3.1 Input File Format

Keywords appear one per each line in the input file and starts with $ sign. In each line, an integer will appear after the keyword and that is the count (frequency) of the keyword (There is a space between keyword and the integer). You need to increment the keyword by that count. Queries will also appear in the input file and once a query (for most popular keywords) appears, you should append the output to the output file. Query will be an integer number (n) without $ sign in the beginning. You should write the top most (n) keyword to the output file. When "stop" (without $ sign) appears in the input stream, program should end. Following is an example of an input file.

```
$facebook 5
$youtube 3
$facebook 10
$amazon 2
$gmail 4
$weather 2
$facebook 6
$youtube 8
$ebay 2
$news 2
$facebook 12
$youtube 11
$amazon 6
3
$facebook 12
$amazon 2
$stop 3
$playing 4
$gmail 15
$drawing 3
$ebay 12
$netflix 6
$cnn 5
5
stop
```

3.2 Output File Format

Once a query appears, you need to write down the most popular keywords to the output file in descending order. Output for a query should be comma separated list without any new lines. Once the output for a query is finished you should put a new line to write the output for the next query. You should produce all the outputs in the output file named "output_file.txt". Following is the output file for the above input file.

facebook,youtube,amazon
facebook,youtube,gmail,ebay,amazon

4 Instructions to run the program

```
make
java src/keywordcounter file_name
```

5 Implementation Details

5.1 Data Structures Used

- Max Fibonacci heap: To keep track of the frequencies of keywords
- Hash table: Keywords are used as keys for the hash table and value is the object reference to the corresponding node in the Fibonacci heap.

5.2 Program Flow

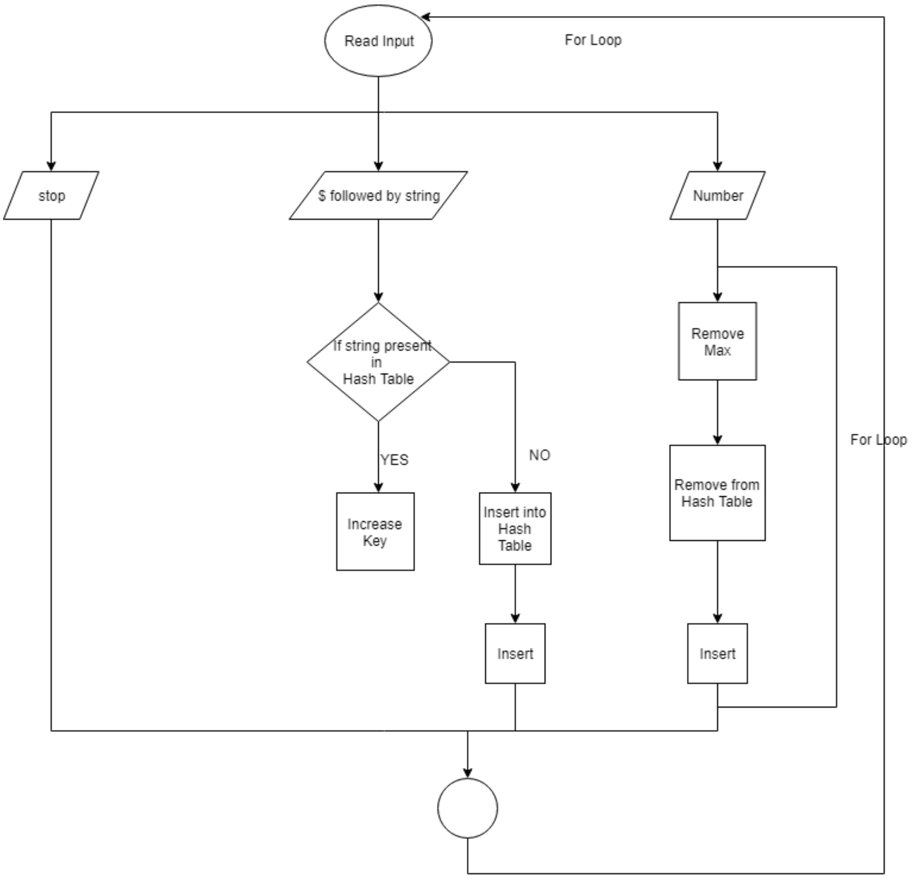
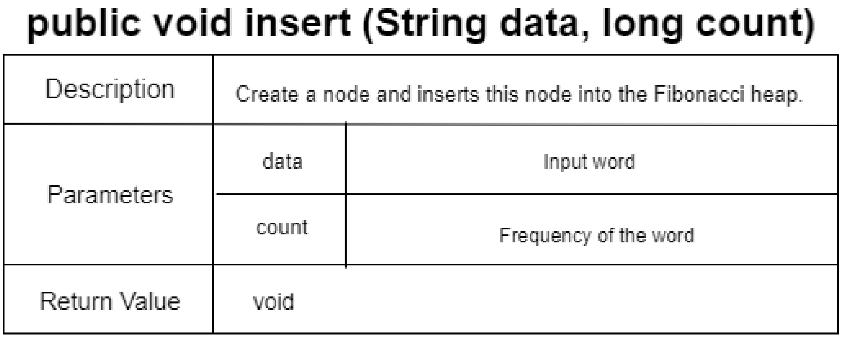


Figure 1: Program Flow

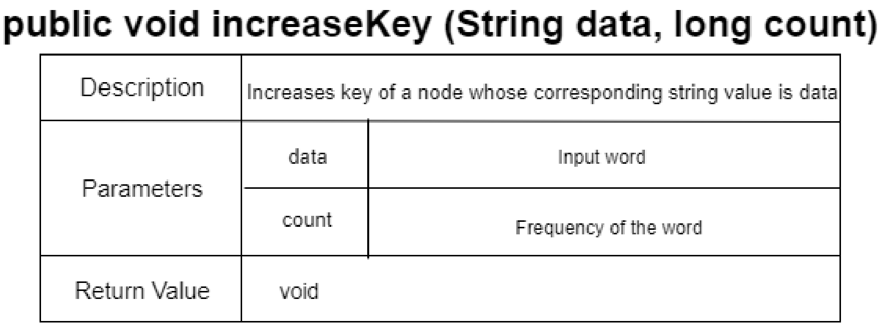
5.3 Function Description



public void insert (String data, long count)

Description	Create a node and inserts this node into the Fibonacci heap.	
Parameters	data	Input word
	count	Frequency of the word
Return Value	void	

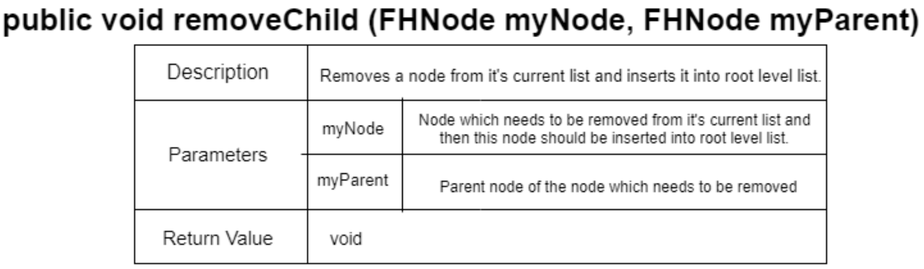
Figure 2: Insert Operation of Fibonacci Heap



public void increaseKey (String data, long count)

Description	Increases key of a node whose corresponding string value is data	
Parameters	data	Input word
	count	Frequency of the word
Return Value	void	

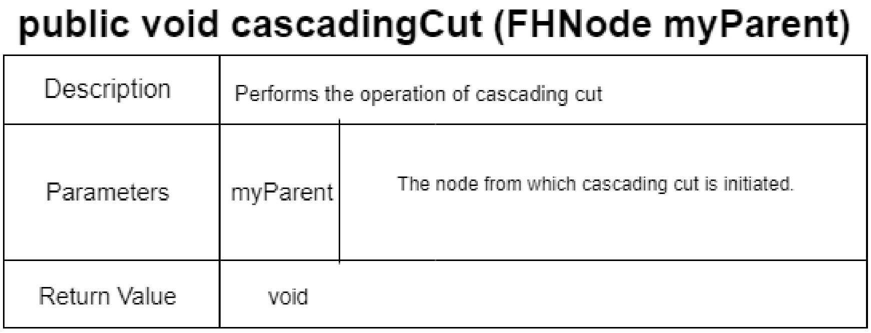
Figure 3: Increase Key Operation of Fibonacci Heap



public void removeChild (FHNode myNode, FHNode myParent)

Description	Removes a node from it's current list and inserts it into root level list.	
Parameters	myNode	Node which needs to be removed from it's current list and then this node should be inserted into root level list.
	myParent	Parent node of the node which needs to be removed
Return Value	void	

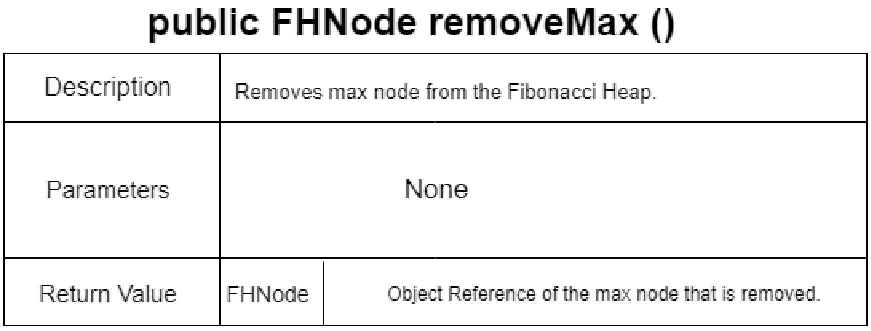
Figure 4: Cut Operation of Fibonacci Heap



public void cascadingCut (FHNode myParent)

Description	Performs the operation of cascading cut	
Parameters	myParent	The node from which cascading cut is initiated.
Return Value	void	

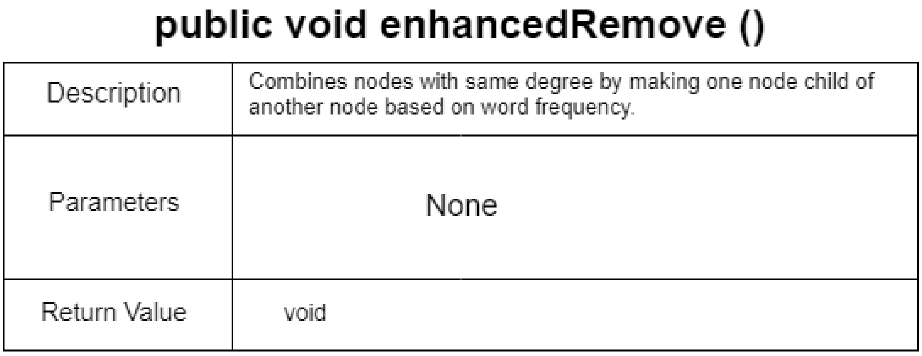
Figure 5: Cascading Cut Operation of Fibonacci Heap



public FHNode removeMax ()

Description	Removes max node from the Fibonacci Heap.	
Parameters	None	
Return Value	FHNode	Object Reference of the max node that is removed.

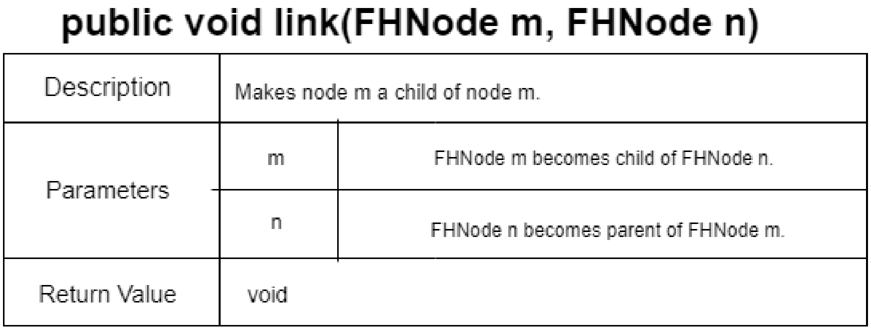
Figure 6: Remove Max Operation of Fibonacci Heap



public void enhancedRemove ()

Description	Combines nodes with same degree by making one node child of another node based on word frequency.
Parameters	None
Return Value	void

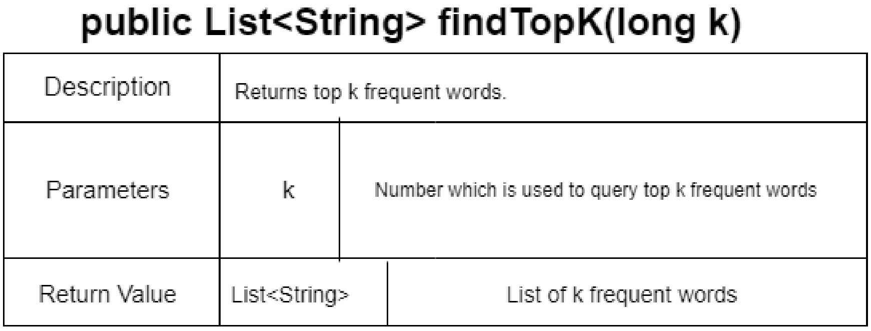
Figure 7: Pairwise Combine Operation of Fibonacci Heap



public void link(FHNode m, FHNode n)

Description	Makes node m a child of node m.	
Parameters	m	FHNode m becomes child of FHNode n.
	n	FHNode n becomes parent of FHNode m.
Return Value	void	

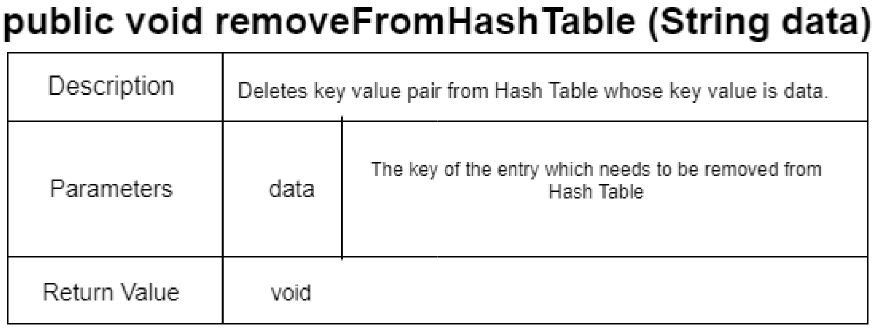
Figure 8: Link Operation of Fibonacci Heap



public List<String> findTopK(long k)

Description	Returns top k frequent words.	
Parameters	k	Number which is used to query top k frequent words
Return Value	List<String>	List of k frequent words

Figure 9: findTopK Operation of Fibonacci Heap



public void removeFromHashTable (String data)

Description	Deletes key value pair from Hash Table whose key value is data.	
Parameters	data	The key of the entry which needs to be removed from Hash Table
Return Value	void	

Figure 10: Remove From Hash Table Operation of Fibonacci Heap