

ADS Project

Name: Sourabh Gopal Parvatikar

UFID:7932-5142

Email id: sourabh.gopalpar@ufl.edu

Function Prototypes:

`void main(String[] args)`

Description	Reads the input file, creates the nodes, increases the node value if necessary, runs a query accordingly if necessary, exits if required and writes the output to the "output_file.txt" file.	
Parameters	args	Contains the name of the input file
Return value	Does not return any value	

`Node(String key, int data)`

Description	Constructor for the node object	
Parameters	key Data	keyword Number of times the keyword has occurred
Return value	Does not return any value	

`void insert(Node node);`

Description	Inserts a node to the max fibonacci heap	
Parameters	Node node	node to be inserted in max fibonacci heap
Return value	Does not return any value	

`void Node removeMax();`

Description	Removes and returns the max node from fibonacci heap	
Parameters	No parameters	
Return value	Max node of type Node	

`void findNextMax();`

Description	Called by removeMax() to find and set the next maximum node	
Parameters	No parameters	
Return value	Does not return any value	

`void pairwiseCombine();`

Description	Performs pair wise combine operation after the delete max. Called by deleteMax() method after it has deleted the max node	
Parameters	No parameters	
Return value	Does not return any value	

`public void makeChild(Node parent, Node child);`

Description	Makes one node the child of another	
Parameters	Node parent Node child	Node to which another node is to be added Node which should become child of parent
Return value		

`public void increaseKey(Node node, int data)`

Description	Inserts a node to the max fibonacci heap	
Parameters	Node node Int data	node whose data field needs to be increased Amount by which data needs to be increased
	Does not return any value	

`public void cut(Node parent, Node child)`

Description	Performs cut and cascading cut operations	
Parameters	Node parent Node child	parent of the node which needs to be cut node which needs to be cut
	Does not return any value	

Workflow:

- The main function in keywordcounter.java reads the input file name, opens it and creates a write file called "output_file.txt".
- If a line starts with \$, it reads the string as an input, else as a query.
- If the keyword in input already exists, it calls increaseKey for that keyword, else it creates a new Node object and calls insert.
- If it is a query, it tries to parse it as an integer. If an exception occurs, it closes the files and exits else it calls removeMax for the number of times mentioned in the query. It inserts the obtained max again into the Fibonacci Heap after writing them to the output file.