Title: Application of Web Crawler and Web Indexing

Data structure used:Graphs, HashMap
Group Members: Tanushree Thakare (2460)
Anaya Tulpule (2462)
Varaa Kukreti (2463)

Details of Designing:

- Classes:

Crawler: This class is for implementing the web crawler.Crawler starts with a seed website (or a wide range of popular URLs (also known as the frontier) and searches in depth and width for hyperlinks to extract. It has the following functions:

1) void bfs() : Breadth First Search Algorithm is used as acrawling algorithm. It starts with the root node and keeps on traversing all the connected nodes to root node. In simple words it starts with one link and keeps traversing all the connected links storing the words present in the title of the page and add all the URLs to an ArrayList. These URLs are then stored in a HashMap with the words present in their title being their key.

2) void display() : Displays the HashMap ie. all the words (which acts as the key) and all the URLs browsed that were associated with that word.

3) ArrayList<String>search(String) : To search and return the ArrayList of URLs associated with the word/s passed as a String to the function.

SearchBox : This class extends the JFrame class and is used to write a GUI application .The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.In its constructor we set the properties/dimensions of all the components that we want to be a part of the search box. It has the following functions:

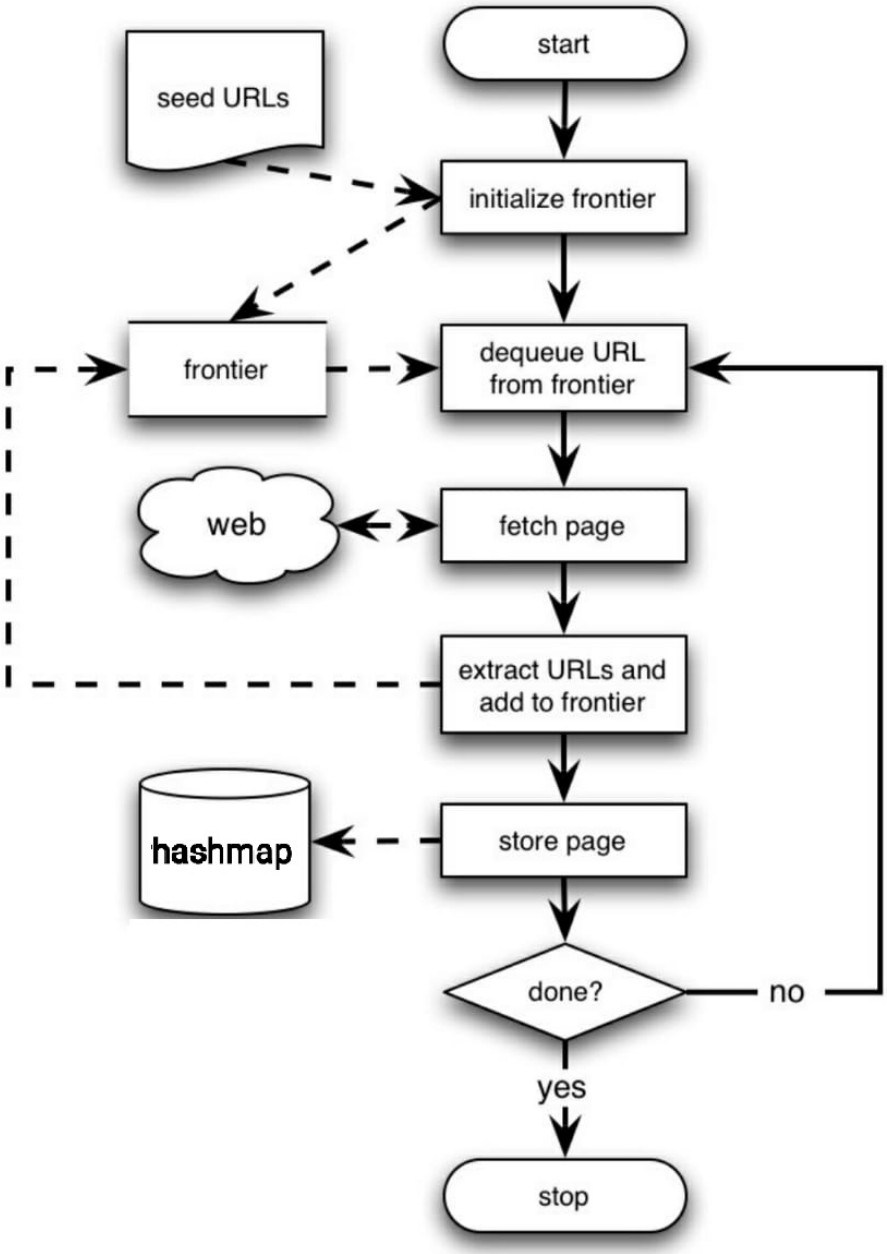
1) void addComponents() : To add the components of the search box to the panel that will be holding them.

2) void setTable() : To add an action listener to the button. When the action is performed (clicking) the steps to be taken are defined in

actionPerformed(ActionEvent e). getText() returns the text contained in this Text Field , which is then passed to displayLabel(String s,Crawlercw), to display results.

3) void displayLabel(String ,Crawler) : String passed to this function is split into words and searched in the HashMap . URLs associated with given words are displayed in text area.

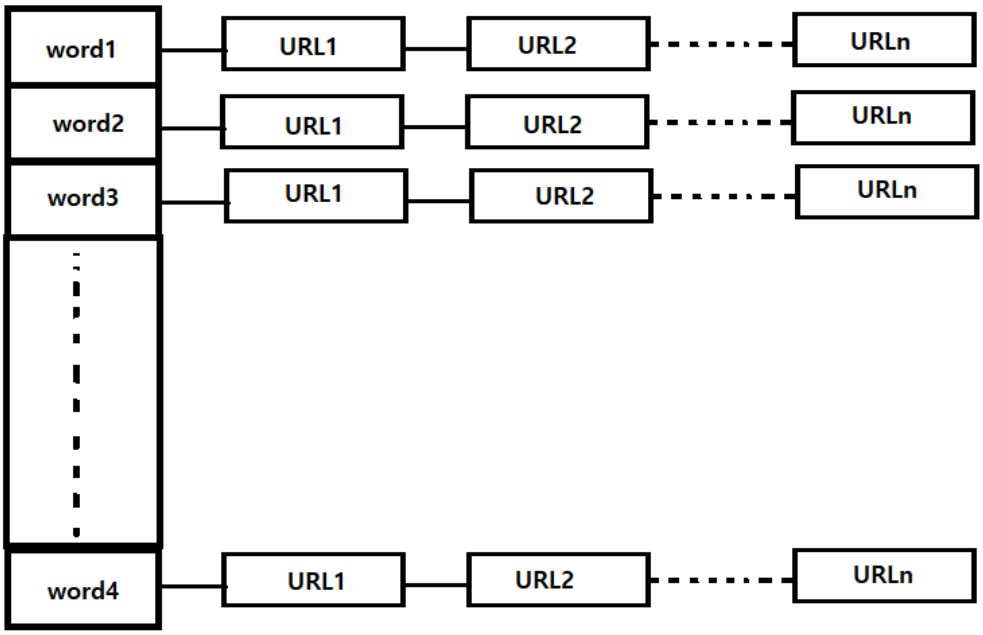
Block Diagram:



Data structure used: Graphs, HashMap :

Bread First Search algorithm was used for browsing the URLS.
Breadth first search is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighbouring nodes. Then, it selects the nearest node and explore all the unexplored nodes. The algorithm follows the same process for each of the nearest node until it finds the goal.

HashMaps use the hashing technique to store the (key , values) pair. Here the words are the Keys and the URLs are the values.The use of a hash table would speeds up execution significantly as its time complexity for searching is O(1). Hash tables and functions support one of the most effective types of searches: *hashing*.



HashMap

Operations Performed :

Logic related to the operations:

void bfs():

It uses regex(Regular expression) to search for hyperlinks on a particular page.

Starting from the seed page we store the links in the queue.
We loop till we haven't reached count = 0 or the queue isn't empty.
For a particular iteration we dequeue a hyperlink from the queue and get that page.We read this page's entire contents as a string,find out hyperlinks and see if it is already there in the visited HashSet if not,add them to visited and store them in queue.
We, also find out using regex the title of a particular page split the words and store them and the link (if not already added) in HashMap of index which maps a particular word to the urls who's title contain that word.
But, if the hyperlink is null we decrement the count and if queue is not empty we add a null link.

void display():

We loop over the keys of the HashMap index print it and get the ArrayList of string associated with it. In the nested loop we print elements of the ArrayList as well.

ArrayList<String> search(String search):

This method has one parameter entered by the user i.e. the string to be searched.
We create a new ArrayList of string to store the resultant urls.
We iterate over each word of the search string and if the word is signifiacnt i.e. it's length is greater than 2 (to avoid words like "of", "an", "if")
and our hashmap contains this word then we obtain its corresponding ArrayList.
We loop through this array list and check if a particular url has already been added to the result. If yes, we skip it and don't add it again else we added it to the ArrayList of result.

void addComponents() :

This method uses panel.add() method to add the components including search button, text field and the text area which is used to display the search results to the panel created. The add() method imported from the Container class is used to add the panel to the frame.

void setTable(): The action listener is added to the button which takes the object of Action listener interface as the parameter. The action listener interface has a predefined method action performed which is passed with the object of Action event class. When the Action is performed i.e. on mouse click the function will be executed. We obtain the text inputted by the user in the search field by get text() function and store it in variable str. The inputted query is then converted

to lower case so that the search will be case insensitive. Then we call the display label function.

void displayLabel(String ,Crawler) :This method is used to display search results in the text field. The Array List r is used to store all the urls related to the search string returned by the search function. If there are no urls related to the string searched i.e. size of array list is zero then we display no search results found. Otherwise we concatenate all the urls and then display them in the Text field by using the setText() method.