

```

331     cout << "3. Exit" << endl;
332     cout << "Type in your choice: ";
333     enter:cin >> x;
334     if (x == 2)
335     |     cont = 1;
336     else if (x == 3)
337     |     exit(0);
338     else if (x == 1){
339     |     cont = 0;
340     |     goto chooseWebpage;
341     | }
342     else{
343     |     cout << "Please enter a valid choice:";
344     |     goto enter;
345     | }
346     }while (cont);
347
348     chooseWebpage:cout << "Which webpage you want to view: ";
349     cin >> numberChoice;
350     if(numberChoice == 1)
351     |     goto choiceDone;
352
353     else if (numberChoice == 2)
354     |     goto choiceDone;
355
356     else

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2. Games|Strategy
<http://www.games.com/strategy>

3. Words
<http://www.words.net>

Would you like to:
 1. Choose a Webpage to open
 2. New Search

SEARCH ENGINE C++

USING PAGE RANK AND CTR IMPLEMENTATION

MOHAMMED ZAIEDA
 AMERICAN UNIVERSITY IN CAIRO
 900181473, Analysis of Algorithms Lab

1. Introduction:

For this project I will build the search engine using a linked list and normal object-oriented programming. I will also use sorting algorithms to optimize the search results.

The reading process will be accomplished using three types of files. One file with keywords and their corresponding weblinks. Another file with links that are connected using graphs. Another file will contain the title of the page link along with the updated ctr value whenever it is accessed by the program.

The indexing of the pages will be in a sorted linked list based on rank numbers of each page. Moreover, the page rank will also be supported by the CTR value stored in the text files which increases the chances of the page being displayed first. The CTR value is updated whenever the user decides to pick a website to view.

In order to compile the program and run it, follow the steps found in the readme file, note that it is required to be built in a **Linux terminal environment for it to completely work, normal compilers might not work for this program.**

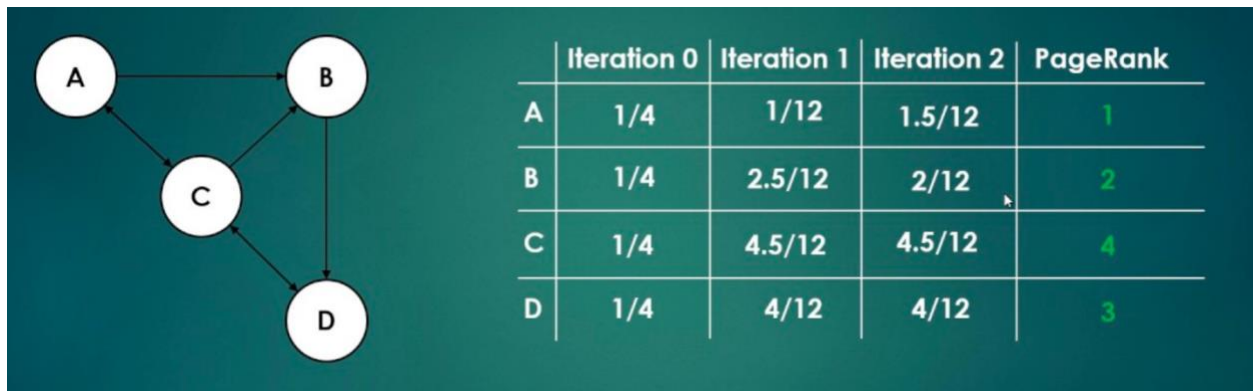
2. Pseudo Code:

For the Page Rank Algorithm:

Using the below formula, I was able to perform the logic in code:

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{PR_t(P_j)}{C(P_j)}$$

Ex:



```

ALGORITHM PageRank(int num, string weblink, int CTR, page*
node) {
    for (int i = 0; i < num; i++)
        for (int j = 0; j < pageNum; j++){
            int x = node -> webLinkSearch(getPageLink());
            for (int k = 0; k < x; k++)
                node->setRanking(node->getPageCTR());
        }

    for (int i = 0; i < num; i++)
        node->setRanking(1.0 / num);

double *arrayPtr = new double[num];
    for (int z = 0; z < 3; z++){
        for (int i = 0; i < num; i++){
            vector<int>* ptr = new vector<int>;
            for (int a = 0; a < node->size(); a++)
                ptr->push_back(node->getRanking()->at(a));

            int ptrSize = ptr->size();
            for (int n = 0; n < size; n++){
                ptr->pop_back();
            }
            arrayPtr[i] = node->getRanking();
        }
        for (int i = 0; i < num; i++)
            node->setRank(arrayPtr[i]);
    }
}

```

Note that the value of CTR is update in the file accessing function implemented below, the value itself is stored in the first line of the text files whenever updated:

```

ALGORITHM read(int num, page *node) {
int CTR = 0;
    for (int i = 0; i < num; i++){
        node->insertPage(i);
        f.open(to_string(i) + "text.txt");
        f >> str;
        node->setPageCTR(i);
        f >> str;
        node->setPageLink(str);
        f >> str;
        node-> setpageTitle(str);
        f.close();
    }
}

```

```

        f.open(to_string(i) + "KeySearch.txt");
        while (!f.eof()) {
            f >> str;
            node-> setKeys(str);
        }
        f.close();

        f.open(to_string(i) + "webLink.txt");
        while (!f.eof()) {
            f >> str;
            node->getPageAt(i)->setWebPages(str);
            CTR++;
        }
        f.close();
    }

```

Return CTR;

Before I perform the page rank algorithm I will sort the nodes of the linked list based on their page rank, such that the page rank algorithm works properly. The sort was done using quick sort algorithm.

```

ALGORITHM sort(){
For(int l = 0; l != swapped; l++){
    swap = 0;
    for (int i = 1; i < getLength(); i++)
    {
        if (node(i)->getRank() < node(i + 1)->getRank())
        {
            if (i == 0)
            {
                ptr = pagePtr;
                pagePtr = ptr->getNextPage();
                ptr->setNextPage(pagePtr->getNextPage());
                pagePtr->setNextPage(ptr);
            }
            Else{
                ptr = node->getNextPage();
                node(i - 1)->setNextPage(ptr->getNextPage());
                getPageAt(i)->setNextPage(ptr);
            }
            swapped = 1;
        }
    }
}
}

```

3. To analyze the time complexity and space complexity the following will be done:

Let n be the number of nodes in the linked list, and m

$$T(n) = T_{\text{sort}}(n) + T_{\text{index in linked list}}(m) + T_{\text{nodes in page}}(n)$$

$$T(n, m) = \sum_{i=0}^n 12 + \sum_{i=0}^m 6 + \sum_{i=0}^4 \sum_{j=0}^n 5$$

$$T(n, m) = \sum_{i=0}^n 12 + \sum_{i=0}^m 6 + \sum_{i=0}^4 \sum_{j=0}^n 5$$

$$T(n, m) = 12 + \sum_{i=1}^n 12 + 6 + \sum_{i=1}^m 6 + \sum_{i=0}^4 \sum_{j=0}^n 5$$

$$T(n, m) = 12(n+1) + 6(m+1) + \sum_{i=0}^4 5(n+1)$$

$$T(n, m) = 12(n+1) + 6(m+1) + 20(n+1) + 4$$

$$T(n, m) = 12n + 12 + 6m + 6 + 20n + 24$$

$$T(n, m) = 32n + 6m + 32$$

$$\therefore T(n, m) = O(n + m)$$

The space complexity is $O(n + m)$.

4. The main data structures used in this project are as follows:

1. Class object-oriented programming
2. Linked List
3. Quick Sorting algorithm
4. Page Rank with CTR updating.
5. Vectors and file streams

Conclusion:

The program could perform the page rank algorithm and click through rate to update the values which allows for more hierarchy in the search algorithm. The search engine works perfectly but requires carefulness when compiling and running to get accurate results. It is also important to note that I assumed that for any search key entered the program automatically recognizes that it should be included in the results, and no need for the quotations characters. For and key perform the following:

Simulation_and_boys, as a search key.

Simulation_or_boys, as a search key,

Sample program:

Search key: simulation

```
Mohammeds-MBP:Algo_project zaieda$ g++ --std c++11 Search_Engine.cpp
Mohammeds-MBP:Algo_project zaieda$ ./a.out
Welcome!
What would you like to do?
1. New Search
2. Exit
Type in your choice: 1
Enter your search key: simulation

1. Words
http://www.words.net/simulation_generate

2. Simulation
https://www.simulation.com/cars

3. simulation_free_online
http://www.simulation_free.com/games_simulation

4. Cards_simulation
http://www.simualtion.com/cards

Would you like to:
1. Choose a Webpage to open
2. New Search
3. Exit
Type in your choice: 1
Which webpage you want to view: 3
You're now viewing webpage 3.
Number of CTR is updated in the info text files corresponding to the searchEngine choice.
Would you like to:
1. New Search
2. Exit
2
Mohammeds-MBP:Algo_project zaieda$
```

Search Key: simulation_and_boys

```
Mohammeds-MBP:Algo_project zaieda$ g++ --std c++11 Search_Engine.cpp
Mohammeds-MBP:Algo_project zaieda$ ./a.out
Welcome!
What would you like to do?
1. New Search
2. Exit
Type in your choice: 1
Enter your search key: simulation_and_boys

1. Words
http://www.simulation.net/simulation_for_boys_games

Would you like to:
1. Choose a Webpage to open
2. New Search
3. Exit
Type in your choice: 1
Which webpage you want to view: 1
You're now viewing webpage 1.
Number of CTR is updated in the info text files corresponding to the searchEngine choice.
Would you like to:
1. New Search
2. Exit
2
Mohammeds-MBP:Algo_project zaieda$
```

Search Key: simulation_or_boys

```
Mohammeds-MBP:Algo_project zaieda$ g++ --std c++11 Search_Engine.cpp
Mohammeds-MBP:Algo_project zaieda$ ./a.out
Welcome!
What would you like to do?
1. New Search
2. Exit
Type in your choice: 1
Enter your search key: simulation_or_boys

1. Words
http://www.simulation.net/simulation_for_boys_games

2. Simulation
https://www.simulation.com/cars_boys

Would you like to:
1. Choose a Webpage to open
2. New Search
3. Exit
Type in your choice: 1
Which webpage you want to view: 1
You're now viewing webpage 1.
Number of CTR is updated in the info text files corresponding to the searchEngine choice.
Would you like to:
1. New Search
2. Exit
2
Mohammeds-MBP:Algo_project zaieda$
```

Search key: gifts

(Error: because it is not added nowhere in the key files found in the directory)

```
Mohammeds-MBP:Algo_project zaieda$ g++ --std c++11 Search_Engine.cpp
^[[AMohammeds-MBP:Algo_project zaieda$ ./a.out
Welcome!
What would you like to do?
1. New Search
2. Exit
Type in your choice: 1
Enter your search key: gifts

Your search key must be in one of the search documents of this project's repository. Try to enter: boys.

Would you like to:
1. Choose a Webpage to open
2. New Search
3. Exit
Type in your choice: 3
Mohammeds-MBP:Algo_project zaieda$
```

Search Key: boy

```
Mohammeds-MBP:Algo_project zaieda$ g++ --std c++11 Search_Engine.cpp
Mohammeds-MBP:Algo_project zaieda$ ./a.out
Welcome!
What would you like to do?
1. New Search
2. Exit
Type in your choice: 1
Enter your search key: boy

1. Words
http://www.simulation.net/simulation_for_boys_games

2. Search_boys
http://www.list.com/boys

3. Listing_all_koko
http://www.koko.com/action

Would you like to:
1. Choose a Webpage to open
2. New Search
3. Exit
Type in your choice: 1
Which webpage you want to view: 2
You're now viewing webpage 2.
Number of CTR is updated in the info text files corresponding to the searchEngine choice.
Would you like to:
1. New Search
2. Exit
2
Mohammeds-MBP:Algo_project zaieda$
```