**REPORT**

**Project 3: Pattern Matching Algorithms**

**Submitted By**

Abdullah Al Raqibul Islam (ID# 801151189)

**Project Objective**

In this project …

**Code Repository Overview**

I have used C++ as the programming language for this project. The project structure looks like this

|  |
| --- |
|  |

Here I am going to give brief description of the purpose of the sub-directories and files of this project:

* /build directory contains the executable program files
* /input directory contains the sample graph inputs
* Makefile will help compiling the project programs and place output executable files in the /build directory
* /problem\_statement holds the problem statement of this project
* README.md contains instructions to build the project and way of running the programs to test further
* /report contains the project reports and helper files for it
* /src holds the programs to solve the problems of this project

**Note**

For runtime analysis I have used standard notations, i.e. B will represent number of nodes and **E** will represent number of edges.

**Project Run Instruction**

|  |
| --- |
| # go to project directory  $ cd project\_2  # build the project  $ make  # command to run solution for problem 1  # general instruction: $ ./build/sss\_path < input\_file > output\_file  # here is a sample  $ ./build/sss\_path < input/sample.in > sample\_sssp.out  # command to run solution for problem 2  # general instruction: $ ./build/mst < input\_file > output\_file  # here is a sample  $ ./build/mst < input/sample.in > sample\_mst.out |

**Algorithm 1: Brute-Force Algorithm**

**Short description:** In this task, I have implemented brute force algorithm for matching pattern in a text.

**Data-structure:** In my solution,I have used …

**Runtime analysis:** The brute force algorithm runs in **O(nm)**.

**Sample Input Set**

Here is the list of my sample input set,

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |

**Algorithm 2: Boyer-Moor-Horspool Algorithm**

**Short description:** For this task, I have implemented Boyer-Moor-Horspool algorithm for matching pattern in a text.

**Data-structure:** I have used …

**Runtime analysis:** The Boyer-Moor-Horspool algorithm’s preprocessing requires **O(S+m)**. In worst case, this algorithm runs in **O(nm)**, and in best cases it runs in **O(n/m)**.

**Algorithm Performance Analysis (w.r.t. comparison required):**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Brute-Force Algorithm |  |  |  |  |  |  |  |  |  |  |
| Boyer-Moor-Horspool |  |  |  |  |  |  |  |  |  |  |
| Boyer-Moor |  |  |  |  |  |  |  |  |  |  |
| Finite Automation |  |  |  |  |  |  |  |  |  |  |
| Knuth-Morris-Pratt |  |  |  |  |  |  |  |  |  |  |

**Algorithm 3: Boyer-Moor Algorithm**

**Short description:** In this task, I have implemented brute force algorithm for matching pattern in a text.

**Data-structure:** In my solution,I have used …

**Runtime analysis:** The brute force algorithm runs in **O(nm)**.

**Algorithm 4: Finite Automation for Pattern Matching**

**Short description:** In this task, I have implemented brute force algorithm for matching pattern in a text.

**Data-structure:** In my solution,I have used …

**Runtime analysis:** The brute force algorithm runs in **O(nm)**.

**Algorithm 5: Knuth-Morris-Pratt Algorithm**

**Short description:** In this task, I have implemented brute force algorithm for matching pattern in a text.

**Data-structure:** In my solution,I have used …

**Runtime analysis:** The brute force algorithm runs in **O(nm)**.

**Test platform**

* Processor: Intel(R) Xeon(R) CPU E5-2620 2.00GHz (12 Core)
* Linux version: 5.0.0-27-generic
* g++ version: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0

**References**

* CP-Algorithms (Prefix function): <https://cp-algorithms.com/string/prefix-function.html>
* Boyer-Moore algorithm: [www.iti.fh-flensburg.de/lang/algorithmen/pattern/bmen.htm](http://www.iti.fh-flensburg.de/lang/algorithmen/pattern/bmen.htm)

**Code**

**Implementation of algorithm 1:**

|  |
| --- |
|  |

**Implementation of algorithm 2:**

|  |
| --- |
|  |

**Implementation of algorithm 3:**

|  |
| --- |
|  |

**Implementation of algorithm 4:**

|  |
| --- |
|  |

**Implementation of algorithm 5:**

|  |
| --- |
|  |