Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring , Year: 2023), B.Sc. in CSE (Day)

Problem Solving (BFS, Sorting, Divide and Conquer)

Course Title: Algorithms Lab
Course Code: CSE 206
Section: 213-DC
Students Details

Name	ID
Md Mehedi Hassan Nayeem	213902045

Submission Date: /29/2023
Course Teacher's Name: Md. Sultanul Islam Ovi

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

This project proposal focuses on solving three distinct problem sets using different algorithms: BFS/DFS, Divide and Conquer, and Sorting. Each problem set presents unique challenges that require specific algorithmic approaches for optimal solutions. The goal is to explore the best-case scenarios for each algorithm and develop efficient strategies to tackle the problems at hand.

1.2 Motivation

Through this project, we aim to analyze and implement the best-case scenarios for each of these problem sets. By exploring the strengths and weaknesses of BFS/DFS, Divide and Conquer, and Sorting algorithms, we can gain a deeper understanding of their capabilities and optimize their application to solve specific problems. This exploration will contribute to advancing algorithmic problem-solving skills and enable us to tackle real-world challenges more effectively.

1.3 Problem Definition

1.3.1 Problem Statement

We have ot solve this three problems

Problem 01: Knight Moves:

The problem described is known as the Traveling Knight Problem (TKP). The objective of this problem is to find the shortest closed tour of knight moves on a chessboard that visits each square of a given set of n squares exactly once. In other words, the task is to determine the minimum number of knight moves required to travel from one square (a) to another square (b) on a chessboard.

A knight in chess moves in an L-shaped pattern, consisting of two steps in one direction (horizontally or vertically) and one step in a perpendicular direction. The knight can move to any square on the chessboard as long as it follows this pattern.

Problem 02 :Dropping Balls :

The problem described involves a fully binary tree structure called FBT (fully binary tree) and a set of K balls that are dropped one by one from the root of the tree. Each ball follows a specific path within the tree until it reaches a leaf node.

The FBT is a binary tree in which every non-terminal node (intermediate node) has exactly two child nodes. The tree is structured in such a way that the nodes on each level are filled from left to right. The nodes in the FBT are sequentially numbered starting from 1, with nodes on the first depth level, then nodes on the second depth level, and so on.

When a ball is dropped, it starts at the root node of the FBT and follows a path down the tree until it reaches a leaf node. The ball's path is determined by the values of flags set in each non-terminal node of the tree.

Initially, all flags in the non-terminal nodes are set to false. When a ball visits a non-terminal node, the flag's current value at that node is checked. If the flag is false, the ball will switch the flag's value to true and follow the left subtree of that node to continue moving down the tree. On the other hand, if the flag is true, the ball will switch the flag's value to false and follow the right subtree of that node.

By following these rules, each ball takes a distinct path through the FBT until it reaches a leaf node, where it stops moving. The objective of the problem is likely to determine the specific leaf node that each ball will reach based on the flag values and the structure of the FBT.

Problme 03 : Meta Loopless Sorts:

The problem at hand involves creating a meta-sorter program, which generates standard Pascal programs that are capable of sorting a given set of n numbers. The objective is to write a program that can generate multiple sorting programs, each with specific properties and constraints.

Sorting algorithms play a crucial role in computer science, and understanding and implementing various sorting algorithms is an important part of computer science education. Sorting tasks consume a significant portion of computational resources worldwide. There is a wide range of sorting algorithms, including well-known ones such as Bubble sort, Quicksort, parallel sorting algorithms, and sorting networks.

1.4 Design Goals/Objectives

My goal is to solve those theree problems and understand the core concept of algorithm.By solving those problem my concept of algorithm must be improved.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	**Explain how to address**
P1: Depth of knowledge required	—- To be good data structure .When we solve the quicksort problem then we must need good knowledge recursion
P2: Range of conflicting requirements	What we learn in our lab ,all problems are related what we learnt in our lab
P3: Depth of analysis required	we need to know time complexity for every problems
P4: Familiarity of issues	—-
P5: Extent of applicable codes	—-
P6: Extent of stakeholder involvement and conflicting requirements	—-
P7: Interdependence	—-

1.5 Application

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

Start the section with a general discussion of the project [1] [2] [3].

2.2 Project Details

In this section, you will elaborate on all the details of your project, using subsections if necessary.

2.2.1 Subsection_name

Figure 2.1: Figure name

You can fix the height, width, position, etc., of the figure accordingly.

2.3 Implementation

All the implementation details of your project should be included in this section, along with many subsections.

2.3.1 Subsection_name

This is just a sample subsection. Subsections should be written in detail. Subsections may include the following, in addition to others from your own project.

The workflow

Tools and libraries

Implementation details (with screenshots and programming codes)

Each subsection may also include subsubsections.

2.4 Algorithms

The algorithms and the programming codes in detail should be included .
Pseudo-codes are also encouraged very much to be included in this chapter for your project.

- Bullet points can also be included anywhere in this project report.

Algorithm 1: Sample Algorithm

```
Input: Your Input
Output: Your output
Data: Testing set $x$
$\sum_{i=1}^{\infty} := 0$                          // this is a comment
/* Now this is an if...else conditional loop                  */
if Condition 1 then
    Do something                                  // this is another comment
    if sub-Condition then
        Do a lot
else if Condition 2 then
    Do Otherwise
    /* Now this is a for loop                                  */
    for sequence do
        loop instructions
else
    Do the rest
/* Now this is a While loop                                    */
while Condition do
    Do something
```

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

Discuss the experimental setup and environment installation needed for the simulation of your outcomes.

3.1.1 Subsection

3.1.2 Subsection

3.2 Results Analysis/Testing

Discussion about your various results should be included in this chapter in detail.

3.2.1 Result_portion_1

The results of any specific part of your project can be included using subsections.

3.2.2 Result_portion_2

Each result must include screenshots from your project. In addition to screenshots, graphs should be added accordingly to your project.

3.2.3 Result_portion_3

Each result must have a single paragraph describing your result screenshots or graphs or others. This is a simple discussion of that particular portion/part of your result.

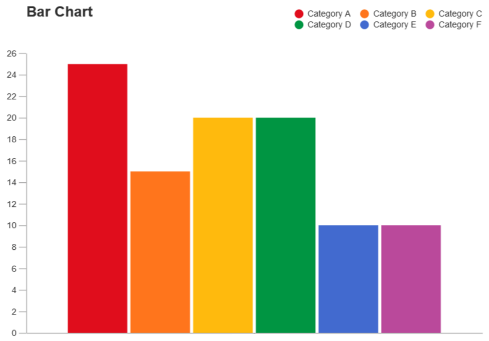

Figure 3.1: A graphical result of your project

3.3 Results Overall Discussion

A general discussion about how your result has arrived should be included in this chapter. Where the problems detected from your results should be included as well.

3.3.1 Complex Engineering Problem Discussion

[OPTIONAL] In this subsection, if you want, you can discuss in details the attributes that have been touched by your project problem in details. This has already been mentioned in the Table **??**.

Chapter 4

Conclusion

4.1 Discussion

Discuss the contents of this chapter and summarized the description of the work and the results and observation. Generally, it should be in one paragraph.

4.2 Limitations

Discuss the limitations of the project. Limitations must be discussed, with the help of some critical analysis.

4.3 Scope of Future Work

Discuss the future work of the project, that is your plans for more work and extension of your project.

References

[1] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.

[2] Douglas Laney. 3d data management: controlling data volume, velocity and variety. gartner, 2001.

[3] MS Windows NT kernel description. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed Date: 2010-09-30.