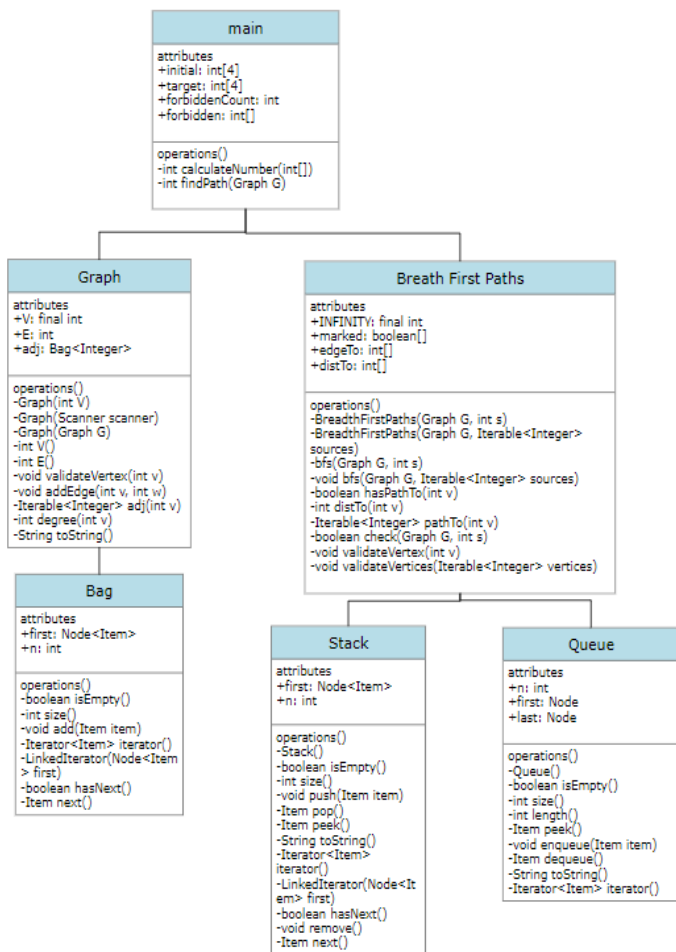


# CMPE343 Programming Homework 2 REPORT

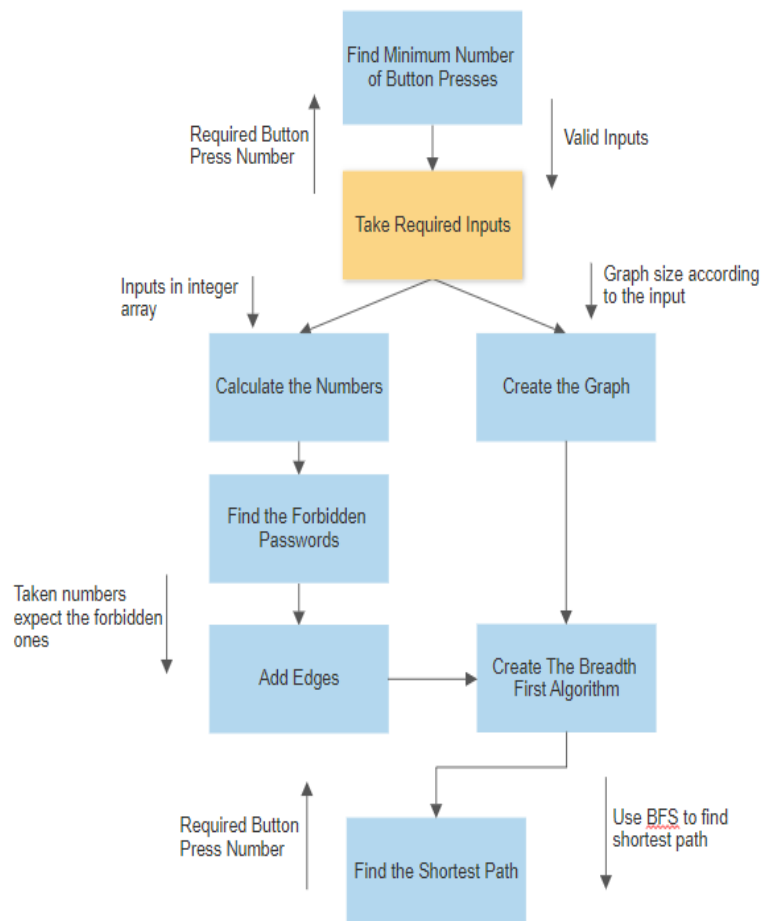
## Problem Statement and Code Design

### Task1

In this task, it is expected to create an algorithm that finds the target password from the initial combination as quick as possible. The program needs to calculate the minimum number of button presses required to transform the initial configuration to the target configuration without passing through a forbidden one. To create the program, I used Graphs to put the number combinations in it and used Breadth First Path algorithm to find the shortest way. The Graph and Breadth First Path have been implemented with the Graph and BreadthFirstPath classes, respectively. Then, in the main class, the input has been taken from the user and put to the according arrays (to convert them to the integer numbers).



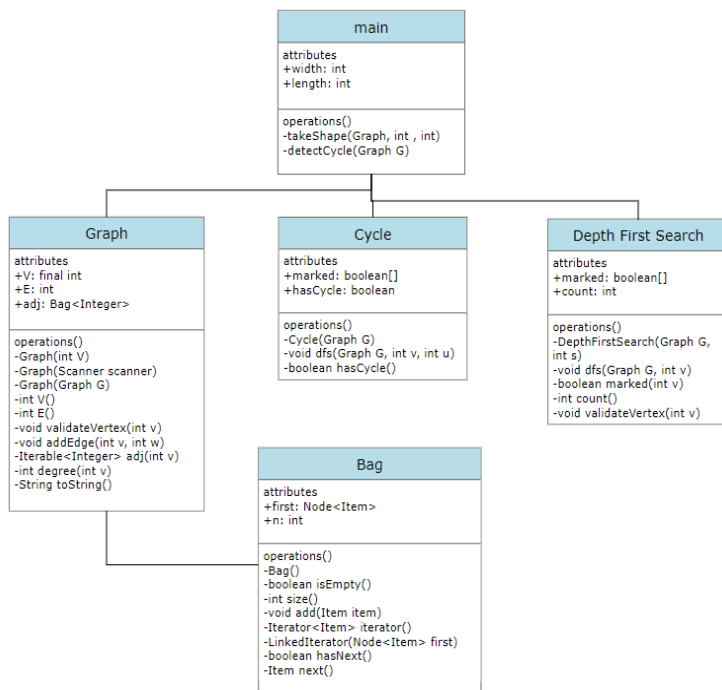
UML Diagram of Task1



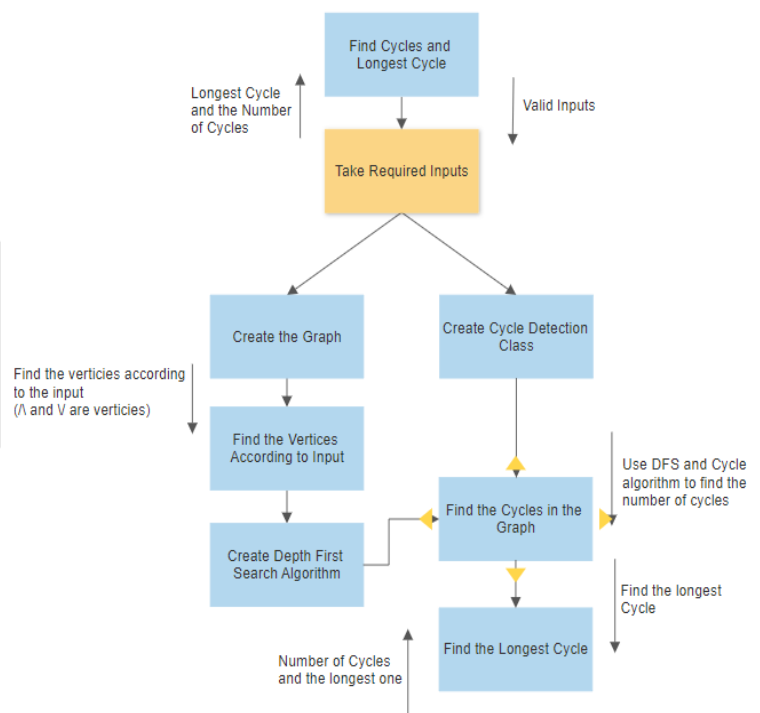
Structure Chart of Task2

## Task2

In the second task, it is expected to create a maze that includes only cyclic paths and paths entering somewhere and leaving somewhere else with the “/” and “\” inputs. Then, it is expected to create a program that counts the cycles in the maze and finds the length of the longest cycle. To create the program and find the cycles, I used Graphs, Depth First Search and Cycle detection algorithms. The Graph, Depth First Search and Cycle algorithms has been implemented with the Graph, DepthFirstSearch and Cycle classes, respectively. Then, in the main class, the information about the maze which is the length and width of the maze has been taken and the slashes (the shape of the maze) has been taken to create the maze.



UML Diagram of Task2



Structure Chart of Task2

## Implementation, Functionality and Performance Comparison

### Task1

In the code, there is 6 different classes. 1 main class and 5 implementation classes. I implemented the breadth first search and graph with the BreadthFirstPaths and Graph classes, respectively. These classes have methods such as Graph, bfs, hasPathTo and distTo. Graph methods are used to create a graph to use. There are 3 different constructor method for Graph class. One of them is designed to take the input and create the graph, other one is to create a graph in given size and the last one is to copy a graph. There is 2 bfs methods, one of them is used to breadth-first search from a single source and other one is used to breadth-first search from multiple sources. hasPathTo method checks if there is a path between the source vertex and given vertex. distTo method finds the number of edges in a shortest path between the source vertex. Then I created the main class, took input from the user and calculated the shortest path and print it. Since there isn't a performance comparison, I did not measure the runtimes.

## Task2

In the code, there is 5 different classes. 1 main class and 4 implementation classes. I implemented the depth first search and graph with the DepthFirstSearch, Graph and Cycle classes, separately. These classes have methods such as Graph, DepthFirstSearch, dfs, marked and hasCycle methods. Graph methods are used to create a graph to use. There are 3 different constructor method for Graph class. One of them is designed to take the input and create the graph, other one is creating a graph in given size and the last one is to copy a graph. DepthFirstSearch method finds the vertices in graph that are connected to the source vertex. dfs method depth-first searches from given vertex. Marked method checks if there is path between the source vertex and target vertex. hasCycle method returns if the Cycle class has found any cycles in the graph. Then I created the main class, took input from the user and found the cycles in it with cycle detection algorithm. Since there isn't a performance comparison, I did not measure the runtimes.

## Testing

### Task1

To test my codes, I used the sample input. The output was:

```
8 0 5 6
The initial code is: 8056
6 5 0 8
The target code is: 6508
5
8 0 5 7
Forbidden code 1 is: 8057
8 0 4 7
Forbidden code 2 is: 8047
5 5 0 8
Forbidden code 3 is: 5508
7 5 0 8
Forbidden code 4 is: 7508
6 4 0 8
Forbidden code 5 is: 6408
The minimum button presses required is: 14
```

### Task2

To test my codes, I used the sample input. The output was:

```
Enter the length of the maze:
6
Enter the width of the maze:
4
Enter the shape of the maze:
\\//\\
\\//\\
//\\//
\\//\\
2 Cycles has been found.
The longest cycle has length 16.
```

## Final Assessments

- The trouble points that I experienced in this assignment was implementing the graphs correctly.
- The implementation and modifying the algorithms parts were the most challenging parts for me because it was hard to think a security code and a slash maze as a graph.
- I learned how to use the breadth-first search and depth first search and how it can be helpful for many tasks.