

**DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING**

**CME1251 PROJECT BASED LEARNING – I
FINAL REPORT
PROJECT – I**

GRAPHER

by

Sadullah CİHAN

Özcan ELMACI

M.Fatih BİRDİR

Lecturers

Prof.Dr. Yalçın ÇEBİ

Res.Asst. Özlem YERLİKAYA

27.12.2019

CHAPTER ONE

PROGRESS DESCRIPTION

The application has a board with 25×40 squares. In this application; user can draw, load, save a graph, and calculate R , R_2 , R_3 , ... , R_n , R^* and R_{min} matrices. R matrix gives directly connected nodes (1 step away). R_2 matrix gives exactly 2 steps away points. R_n matrix gives exactly n steps away points. R^* matrix gives all connected points. R_{min} matrix gives the minimum number of steps required for going point a to point b .

First week:

We managed to we create our communication network with trello and start to learn the details we need to use to create a neat project. We asked a lot of questions to each other and give notification through the all way of building the mainframes and structure of our work. We increase our knowledge about the details and important parts gradually. For instance, using 'Functions'. However, like any other projects we encountered inevitable problems while we were building our algorithm. Even though, it was a hard and slow process. We proceed through our schedule. It was critically important to communicate and build an understandable and readable solution strategy to make our job easy for the coming steps of developing. Especially while writing our code.

Second week:

We start to determining the variables we need to use to make everything readable and clean, we pay attention to naming process because to be not confused we know the fact that we need to use variables in a good coding style. We added some crucial important part of our project to our flowchart like the rules we need to code and like the path we need to build our code on. Then we implamented our code.

Third week:

We started to do some real coding and understand the fact flowcharts are crucially important for further steps in our way to make everything easier to understand.

We help each other so it was way easy to understand each others code. Each of us made the things that assign to us and completed. When we encountered a problem with our code we made research to overcome with it. So, with this was we learned lots of new stuff for c#. “Functions and procedures” were one of the codes we learned in our classes and it was really useful for our work. Also, this week was too tough for all of us. Because we have a midterm week. Hence, we were forced. We practice it a lot and then finally added it to our project with after finding errors and editing.

Fourth week and fifth week:

It was one of the most important parts of our project because it was time to organize everything and create a good appearance for our group. To do it, the work that we did last weeks was really helpful. After completing our codes finally, we were able to show what we have done for the last 5 weeks. To be able to do it we started doing our documentation first. It was a good idea to do it first because after that we were able to shape another part of our presentation around it. Everybody was able to handle their documentation part because everybody knew what they are doing and participate in every aspect of the project. Then, we managed to create an appealing poster which was reaching the criteria.

CHAPTER TWO

TASK SUMMARY

2.1. Completed Tasks

Sadullah Cihan :

We organized the team for a meeting. Moreover, we created the necessary functions and variables for our project. It was very tough for the first week. We thought about how to move the cursor. We learned how to load the graph. We designed the tracing algorithm. Also, we have finished the saving process in the third week. In the fourth week, we needed to form the $R_2, R_3 \dots R_n$ matrices. We have accomplished that with the recursive functions that we learned that week. In the last week, we created R^* and R_{min} matrices. Consequently, we prepared the progress report and final report. Then, we always thought about how we could do our other duties. We made an algorithm for tracing and all matrix with my friends by meeting outside. Finally, we made a poster and this final report.

Özcan Elmacı:

As we said in the progress report, we organized 'trello' for team communication. We planned all the coding processes. Besides, we solved 'drawing manually, saving (however its 4th weeks duty) and loading'. We controlled the 8 ways around the letter and found the '+'. Then we found path changings. Moreover, we tested the code and prepare the progress report and final report. We could not finish tracing properly. We had several problems. But, we handled it and made R_Matrix and the others.

Mehmet Fatih Birdir :

In the first week we try to understand what kind of project we are gonna do, we created an algorithm outline and created an array on the screen. Then we wrote the moving cursor and editing a graph code. After learning the file reading processes, we wrote the code of reading and loading graph file. We were able to show the graph on the screen. In the third week, we needed to handle tracing the graph and form the R matrix. It was hard to do that we tried several ways and then agreed on putting dots on the "+" signs so it can do the tracing easier. Then we formed the R matrix and wrote the progress report. Then my friends completed the rest of the project. Finally, we prepared the Final report, Poster and the Presentation.

2.2. Incomplete Tasks: Reasons and Explanations

There is no incompleted tasks.

2.3. Additional Improvements to the Project

Sadullah Cihan:

We used 'RECURSIVE' function for creating R2, R3.....Rn matrices. You can see in the poster that contains the QR code that contains a recursive matrix multiplication algorithm.

Özcan Elmacı:

We have improved 'RECURSIVE' function. It is very useful.

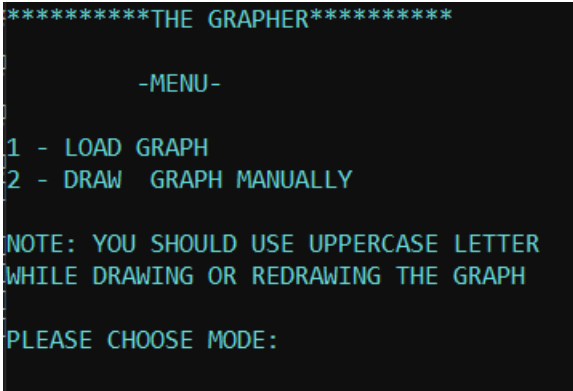
Mehmet Fatih Birdir :

My friends have improved 'RECURSIVE' function. I could not have thought like that algorithm.

CHAPTER THREE

EXPLANATION OF ALGORITHMS

2.4. Screenshots



```
*****THE GRAPHER*****  
  
-MENU-  
  
1 - LOAD GRAPH  
2 - DRAW GRAPH MANUALLY  
  
NOTE: YOU SHOULD USE UPPERCASE LETTER  
WHILE DRAWING OR REDRAWING THE GRAPH  
  
PLEASE CHOOSE MODE:
```

```
Enter the option of which graph you want to see
1-Existed graph
2-New existed graph
```

```

+++
+.+.
EX+.AX+++++++
X.++.+
+.+.+.
+.+.+.B
+.+.+.XX
+.+.+.+.
+.X.+.+.
DX+++++++C+++++++
X+.+.+.
+.+.+.
+.+++++.+.+.
+.+.+.+++++XH
+.+.+.X
+.X.X.+.
FX+++++I.G+++++++

```

MENU

'0' => Show R Min Matrix

'1' => Show R and R * Matrix

'2-9' => Show Rn Matrix

'Q' => Query For Min Steps

'C' => Change The Graph


```
#####
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#####
press enter to save new graph
```

```
#####
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#.....#
#####
Saving operation has finished successfully !
press enter to save new graph
```

2.5. Functions

We used lots of function for our project. Because copying and pasting a part of code is not good every time and extends code. We used functions for tracing, creating matrices. In addition to all these functions, we used 'Recursive Function'. We have used the File.OpenText() function to load the graph. RECURSIVE ALGORITHM:

```
static int[,] R_N(int N)
{
    if (N == 1) return R_N_MATRIX;
    else
    {
        int sum = 0; // MATRIX MULTIPLICATION
        for (int i = 0; i < 16; i++)
        {
            for (int j = 0; j < 16; j++)
            {
                sum = 0;
                for (int k = 0; k < 16; k++)
                {
                    sum += R_N_MATRIX[i, k] * R_ORIGINAL[k, j];
                }
                if (sum > 1)
                { sum = 1; }
                R_N_MATRIX_NEW[i, j] = sum;
            }
        }
        //R_N_MATRIX has been updated here
        for (int i = 0; i < 16; i++)
        {
            for (int j = 0; j < 16; j++)
            {
                R_N_MATRIX[i, j] = R_N_MATRIX_NEW[i, j];
            }
        }
        N--;
        return R_N(N);
    }
}
```

```
static void MENU()...
1 başvuru
static void GRAPH_LOADER()...
1 başvuru
static void GRAPH_DRAWER()...
1 başvuru
static void GRAPH_REDRAWER()...
2 başvuru
static void SAVING()...
static void right()...
1 başvuru
static void right_down()...

1 başvuru
static void right_up()...
1 başvuru
static void up()...
1 başvuru
static void down()...
1 başvuru
static void left()...
1 başvuru
static void left_up()...
1 başvuru
static void left_down()...
static void TRACING_AND_R_MATRIX()...
```

3.1 Algorithms and Solution Strategies

Sadullah Cihan: Firstly, we discussed with each other to create solution strategies. We don't know what to do about this project. Nevertheless, we have done a lot of research. Thus, we learn various functions such as `graph_txt.Close()`; We have used the `File.OpenText()` function to load the graph. Solving the tracing issue was too hard. However, we designed our algorithm. Then we coded as our lecturer said.

In tracing algorithm we have tried the following steps. Firstly, it finds letters wherein the main array. Secondly, it looks around in 8 ways. Then it counts plus amount to learn how many ways of starting in that letter. Then we created another array not to come back. Finally, if it founds 'X', the following letter must be the connected node. We captured that point here. The fundamental tracing algorithm was like that.

Özcan Elmacı: Firstly, we tried to understand this project. We thought to code as always. But our teacher told us what we should do first. We understood that creating a flowchart or pseudocode is the thing we need. We created a flowchart. To read a text file, we used the information we learned from our Algorithm lessons such as, "StreamReader". Also, we learned how to use a function and procedure. Then we started to create an algorithm for our encountered problems. At the beginning of tracing, we made the code followed '+' but it was not true and after that made the code followed until the square is '.'.

We finished tracing and started creating all matrices. Although it was difficult for us, we did it well and we made them showed on the screen whenever users want to see.

Mehmet Fatih Birdir : As first we could not make the cursor movement. We tried to integrate the cursor movement code that we used before but that code was written for one-dimensional array so it was working differently and we needed to change it for a 2-dimensional array because we were working on a 2d array in this Grapher project. After finding the problem which was the coordinates because cursor's x coordinate was matching the arrays y coordinate so we changed it. Another one was with the tracing algorithm. We tried to follow the "+" but it might go to the "+" that it has gone before so we think that if we change the "+" to "." It will not go back and it can do the tracing without any crash.

CHAPTER FOUR

PROBLEMS ENCOUNTERED

Sadullah Cihan:

Firstly, we had a little bit of knowledge about this project. Therefore, we understood clearly. However, we did not know the functions and procedures. Also, we had not enough knowledge about file operations such as loading TXT file, saving a TXT file. The second week, we have learned these subjects. As soon as we learn, we tried to transfer to code.

Of course, we dealt with problems. For instance, we found hidden ASCII characters when we were in the debugging process. Nonetheless, we have overcome that problem. Then we accomplished TXT file loading. The tough part was the tracing part. We designed our algorithm again and again. Sometimes we could not transfer algorithm to code. Consequently, we adapt the algorithm efficiently.

We faced a problem with R* matrix. We did all thing we could do but the code did not work. Because we wrote contents of the cursor such as (writing `Console.SetCursorPosition(i,j)` instead of `Console.SetCursorPosition(j,i)`). Then we fixed this and created all matrices well.

Özcan Elmacı: As always, we encountered a lot of problems. We tried to read a TXT file, but we could not do at the beginning. Then, we did debug and realised that there were several characters at the end of every row like “/r or /t”. After that, we handled it.

We wanted to write our new graph into a TXT file and wrote. But this time we could not make a new graph written. Because, in our second for loop, we had written “Console.WriteLine()” to pass another row, but we added one more. We were warned with this message “Out of index”. After that, we fixed it.

While we were trying tracing and forming the matrix, we faced several problems related to our variables and fixed. We tried to trace normally without using a fake screen. But we could not do that and decided to use a fake screen that works background of the real screen.

Mehmet Fatih Birdir: We have encountered lots of problems through 5 weeks but we handled them after thinking about the possible ways to deal with them. Some of them were on Tracing part, for example, it was not tracing loops like "E" to "E". The other was R* and Rmin because they were hard to code. Another problem was, as I mentioned before, cursor movement. Another was again on the tracing part. It was the problem with deleting the "+" signs because when we do that it also deletes the "+" which was a way of another line. So we make it not to delete it if there are more than 3 "+" signs after seeing it worked.

CHAPTER FIVE

CONCLUSION

In conclusion, we learn how to create a ‘function’ which let us help to simplify something algorithmically.

We have learned to time-share thanks to this project and our school. Because our homeworks ,exams,and project overlapped. Although all these things happened ,we did this project well. Of course,we realized that all project that is given us improves our code skills. We apply what we have learned in algorithm lessons as possible as we can.

REFERENCES

Herbert Schildt, C# 4.0: The Complete Reference, McGraw-Hill, 2010.

Paul Deitel, Harvey Deitel, C# 2010 for Programmers, Pearson Education, 2011.

<https://www.kodevreni.com>

https://www.tutorialspoint.com/csharp/csharp_arrays.htm

<https://docs.microsoft.com/tr-tr/dotnet/csharp/programming-guide/classes-and-structs/local-functions>

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>

<https://www.canva.com>

APPENDIX A

POSTER OF THE PROJECT

GRAPHER

INTRODUCTION

- The application has a board with 25*40 squares.
- In this application; user can draw, load, save a graph, and calculate R, R2, R3, ... , Rn, R* and Rmin matrices.
- R matrix gives directly connected nodes (1 step away).
R2 matrix gives exactly 2 steps away points.
- Rn matrix gives exactly n steps away points. R* matrix gives all connected points.
- Rmin matrix gives the minimum number of steps required for going point a to point b.

WELL DESIGNED CONSOLE INTERFACE

```

R MATRIX          R*
ABCDEFGHIJKLMNPO  ABCDEFGHIJKLMNPO
A 0010000100000000 A 111111111100000000
B 1000000000000000 B 111111111100000000
C 0101001000000000 C 111111111100000000
D 0000100010000000 D 000111001000000000
E 0000100000000000 E 000010000000000000
F 0001000000000000 F 000111001000000000
G 0000000100000000 G 111111111100000000
H 0100000000000000 H 111111111100000000
I 0000010000000000 I 000111001000000000
J 0000000000000000 J 000000000000000000
K 0000000000000000 K 000000000000000000
L 0000000000000000 L 000000000000000000
M 0000000000000000 M 000000000000000000
N 0000000000000000 N 000000000000000000
O 0000000000000000 O 000000000000000000
P 0000000000000000 P 000000000000000000

MENU          QUERY FOR MIN STEPS
'0' => Show R Min Matrix      From: E To: E
'1' => Show R and R * Matrix  1
'2-9' => Show Rn Matrix
'Q' => Query For Min Steps
'C' => Change The Graph
    
```



RECURSIVE MATRIX MULTIPLICATION



2018510016 SADULLAH CİHAN
2018510023 ÖZCAN ELMACI
2018510013 M.FATİH BİRDİR

