

# COP 4534 Project 2 Conclusions

Authors:

- Daniel Gonzalez 6285324 Algorithm Techniques U02
- Muhammad Hashim 6349162 Algorithm Techniques U01

## Conclusions:

- When implementing the solution within the project, Muhammad and I found that there were lots of difficulties trying to graph each vertex and it's edge. Although we had completed a program in class in which we did this before, trying to make it dynamic and creating a class that would dynamically visualize the graph would have likely been the hardest challenge. Muhammad and I learned how to use JFrame a lot more, and also brushed up on our object-oriented programming skills. While creating the Dijstrakas method, we both learned how the algorithm works in more depth.

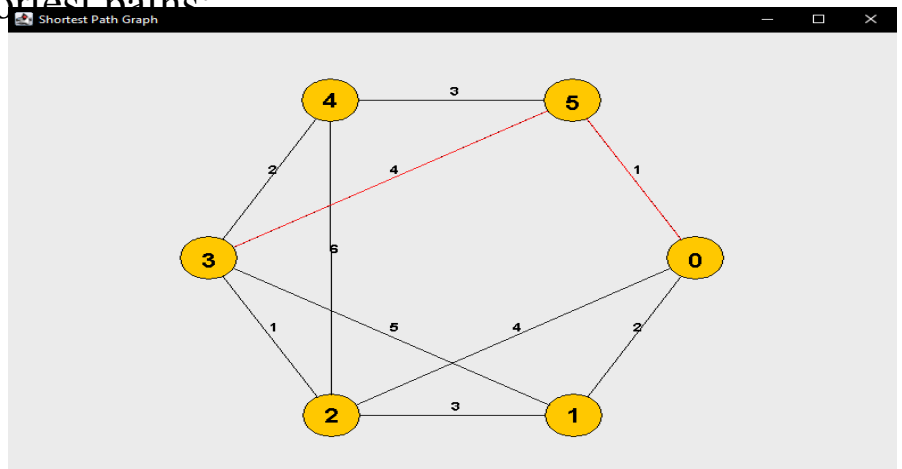
## Visualized graphs & their shortest paths:

- Graph 1 (0 - 3):

- ```

- 6
- 0 2 4 0 0 1
- 2 0 3 5 0 0
- 4 3 0 1 6 0
- 0 5 1 0 2 4
- 0 0 6 2 0 3
- 1 0 0 4 3 0
- 0 3

```

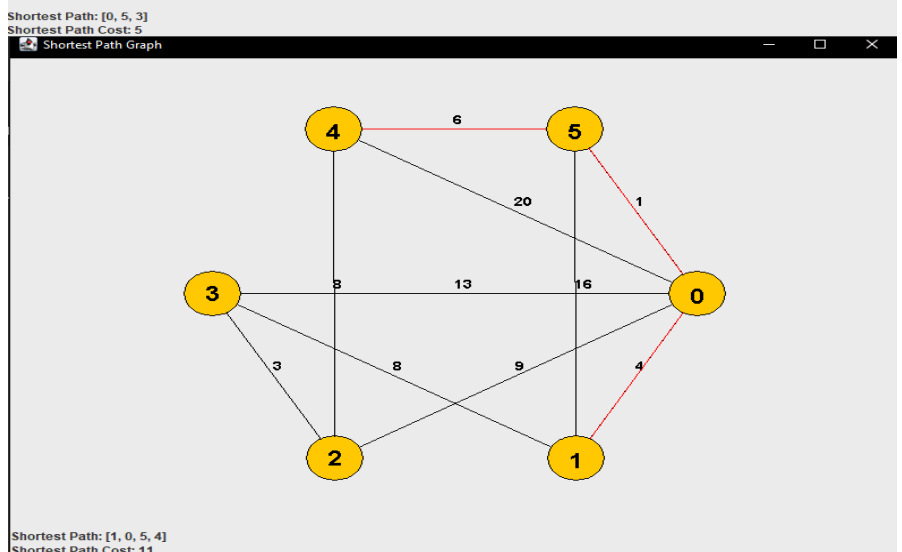


- Graph 2 (1 - 4):

- ```

- 6
- 0 4 9 13 20 1
- 4 0 0 8 0 16
- 9 0 0 3 8 0
- 13 8 3 0 0 0
- 20 0 8 0 0 6
- 1 16 0 0 6 0
- 1 4

```



IDE: IntelliJ IDEA Community Edition 2022.3

Project: COP4534\_Project2

File: ShortestPath.java

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.io.*;
4 import java.io.FileNotFoundException;
5 import java.util.*;
6 import java.util.Scanner;
7
8
9 /**
10  * PrimalityTest Class
11  *
12  * @author
13  * Daniel Gonzalez 6285324 Algorithm Techniques UO2
14  * Muhammad Hashim 6349102 Algorithm Techniques UO1
15  */
16
17 class Graph {
18     private int V;
19     public int[][] graph;
20
21     public Graph(int V) {
22         this.V = V;
23         graph = new int[V][V];
24     }
25
26     /**
27      * Method which adds the edge given the starting vertex, ending vertex, and weight
28      */
29 }

```

Shortest Path Graph

Shortest Path: [0, 5, 3]  
Shortest Path Cost: 5

Run: ShortestPath

```

C:\Users\Hashim\jdk\openjdk-19.0.1\bin\java.exe "-javaagent:E:\Idea\IntelliJ IDEA Community Edition 2022.3\lib\idea_rt.jar=1290:E:\Idea\IntelliJ IDEA Community Edition 2022.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
Shortest path from 0 to 3:
0-5-3

```

IDE: IntelliJ IDEA Community Edition 2022.3

Project: COP4534\_Project2

File: ShortestPath.java

```

119 * to the console the shortest path and displays the shortest path highlighted red as well.
120 *
121 * @return VOID
122 */
123 public void createAndShowGUI() {
124     try {
125         File file = new File("graph_input2.txt");
126         Scanner scanner = new Scanner(file);
127
128         int V = scanner.nextInt();
129         g = new Graph(V);
130
131         for (int i = 0; i < V; i++)
132             for (int j = 0; j < V; j++) {
133                 int weight = scanner.nextInt();
134                 if (weight != 0)
135                     g.addEdge(i, j, weight);
136             }
137
138         int s = scanner.nextInt();
139         int t = scanner.nextInt();
140
141         shortestPath = g.dijkstra(s, t);
142         shortestPathCost = g.getShortestPathCost(s, t);
143
144         System.out.println("Shortest path from " + s + " to " + t + ":");
145         System.out.println(shortestPath);
146         System.out.println("Shortest Path Cost: " + shortestPathCost);
147     } catch (FileNotFoundException e) {
148         e.printStackTrace();
149     }
150 }

```

Shortest Path Graph

Shortest Path: [1, 0, 5, 4]  
Shortest Path Cost: 11

Run: ShortestPath

```

C:\Users\Hashim\jdk\openjdk-19.0.1\bin\java.exe "-javaagent:E:\Idea\IntelliJ IDEA Community Edition 2022.3\lib\idea_rt.jar=1290:E:\Idea\IntelliJ IDEA Community Edition 2022.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
Shortest path from 1 to 4:
1-0-5-4

```