Christine Amelia – 05111740000174

Leonita M. R. S. – 055111740000108

Rimas M. I. – 05111740000185

DAA-F Quiz 2 Report

Group 2

Project: Shortest Path Using Dijkstra

Problem Abstraction

Based on Dijkstra's History: What is the shortest way to travel from Rotterdam to Groningen, in general: from given city to given city. It is the algorithm for the shortest path, which I designed in about twenty minutes. One morning I was shopping in Amsterdam with my young fiancée, and tired, we sat down on the café terrace to drink a cup of coffee and I was just thinking about whether I could do this, and I then designed the algorithm for the shortest path. As I said, it was a twenty-minute invention. In fact, it was published in '59, three years late. The publication is still readable; it is, in fact, quite nice. One of the reasons that it is so nice was that I designed it without pencil and paper. I learned later that one of the advantages of designing without pencil and paper is that you are almost forced to avoid all avoidable complexities. Eventually that algorithm became, to my great amazement, one of the cornerstones of my fame. — *Edsger Dijkstra, in an interview with Philip L. Frana, Communications of the ACM, 2001.*

Design Analysis Algorithms

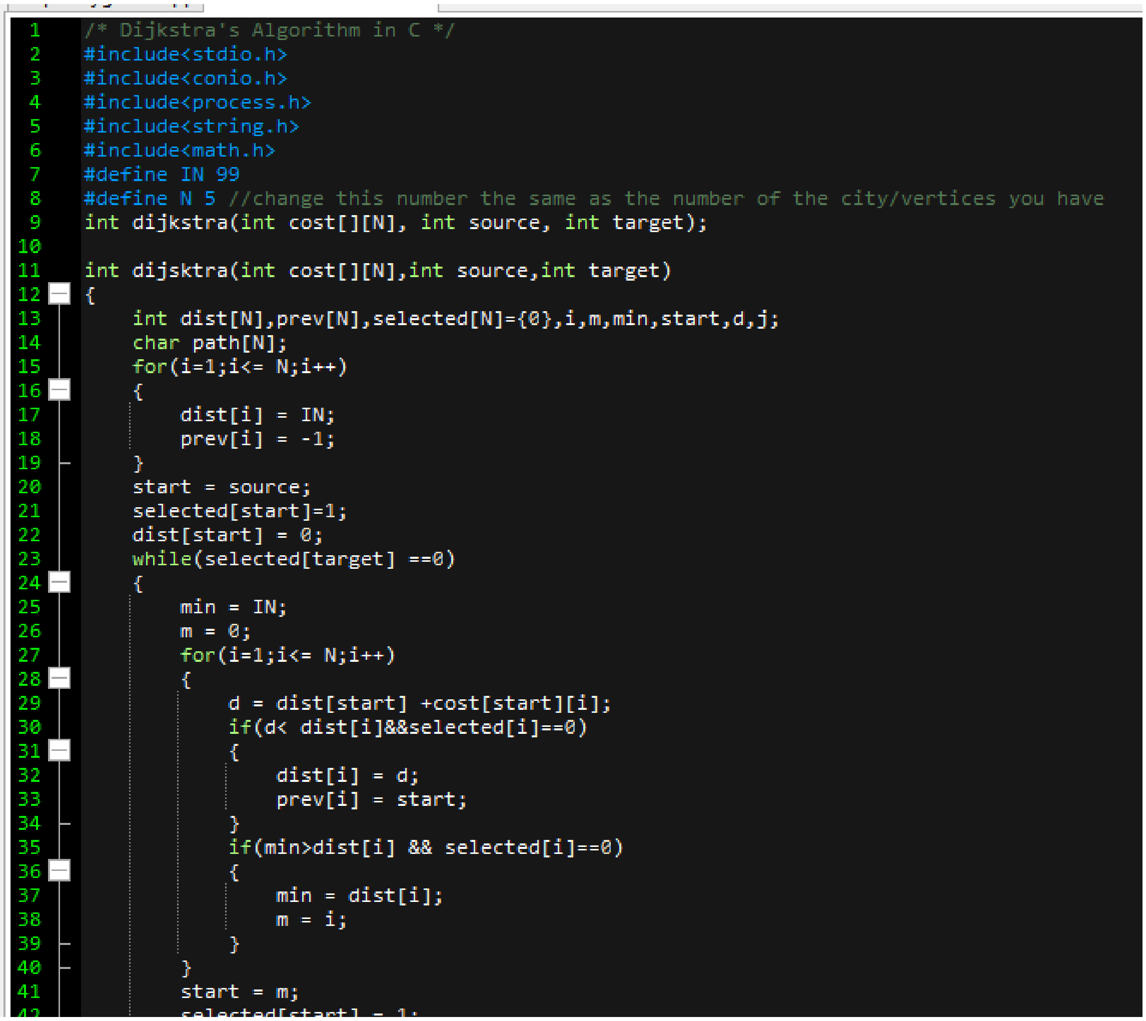
We made this program to find the shortest distance between two cities (vertices). We are using Dijkstra's Algorithm to solve this. At first you will be asked to write down the distance(edge) between two cities as asked by the program and after that you will be asked which one will be the start city(vertex) and which one will be your destination, and then it will compute the shortest distance it could find, and also the path.

Christine Amelia – 05111740000174

Leonita M. R. S. – 055111740000108

Rimas M. I. – 05111740000185

Source Code



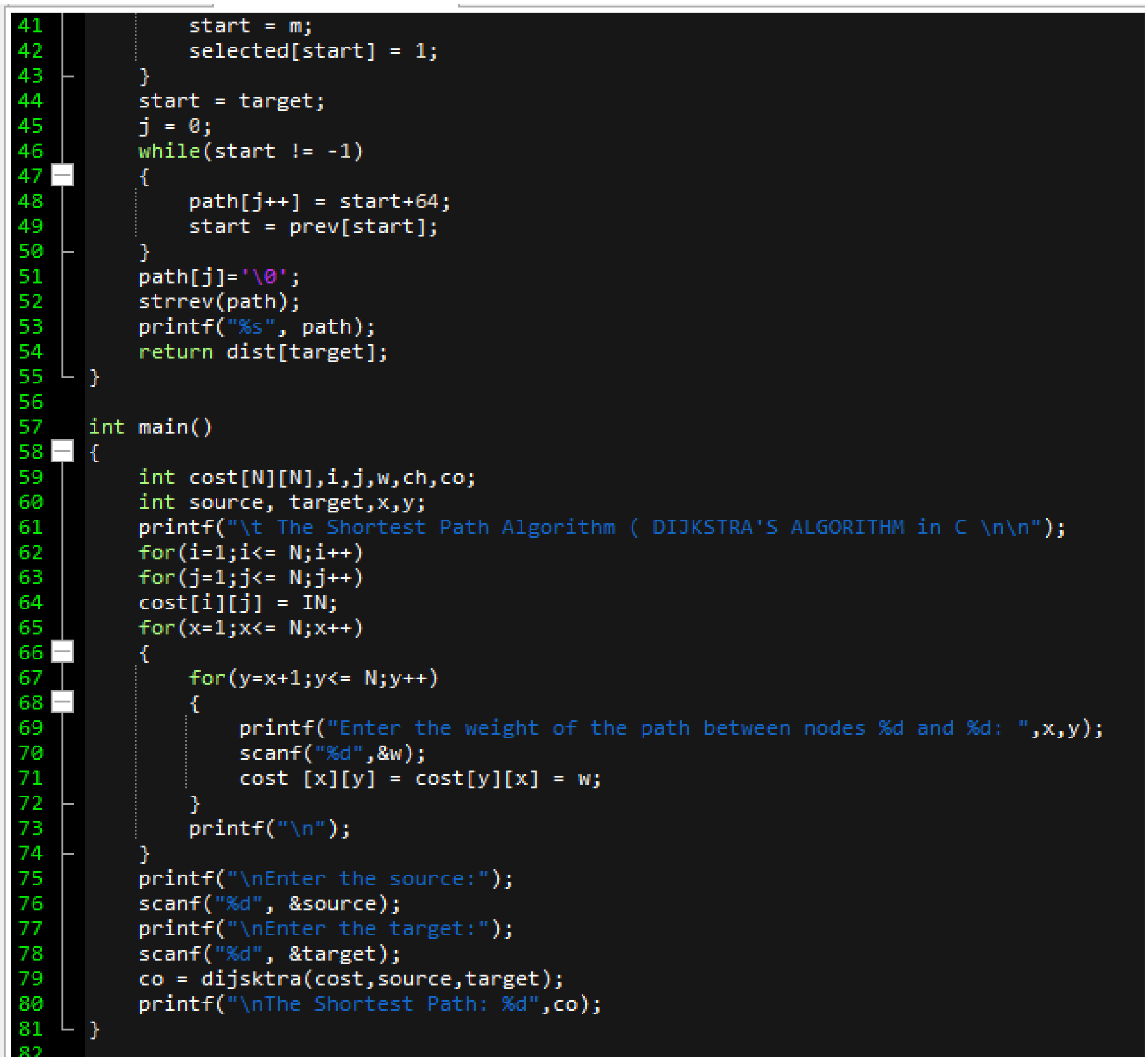
```c
/* Dijkstra's Algorithm in C */
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<string.h>
#include<math.h>
#define IN 99
#define N 5 //change this number the same as the number of the city/vertices you have
int dijkstra(int cost[][N], int source, int target);

int dijsktra(int cost[][N],int source,int target)
{
    int dist[N],prev[N],selected[N]={0},i,m,min,start,d,j;
    char path[N];
    for(i=1;i<= N;i++)
    {
        dist[i] = IN;
        prev[i] = -1;
    }
    start = source;
    selected[start]=1;
    dist[start] = 0;
    while(selected[target] ==0)
    {
        min = IN;
        m = 0;
        for(i=1;i<= N;i++)
        {
            d = dist[start] +cost[start][i];
            if(d< dist[i]&&selected[i]==0)
            {
                dist[i] = d;
                prev[i] = start;
            }
            if(min>dist[i] && selected[i]==0)
            {
                min = dist[i];
                m = i;
            }
        }
        start = m;
        selected[start] = 1;
```

Christine Amelia – 05111740000174

Leonita M. R. S. – 055111740000108

Rimas M. I. – 05111740000185



```
41          start = m;
42          selected[start] = 1;
43      }
44      start = target;
45      j = 0;
46      while(start != -1)
47      {
48          path[j++] = start+64;
49          start = prev[start];
50      }
51      path[j]='\0';
52      strrev(path);
53      printf("%s", path);
54      return dist[target];
55  }
56
57  int main()
58  {
59      int cost[N][N],i,j,w,ch,co;
60      int source, target,x,y;
61      printf("\t The Shortest Path Algorithm ( DIJKSTRA'S ALGORITHM in C \n\n");
62      for(i=1;i<= N;i++)
63      for(j=1;j<= N;j++)
64      cost[i][j] = IN;
65      for(x=1;x<= N;x++)
66      {
67          for(y=x+1;y<= N;y++)
68          {
69              printf("Enter the weight of the path between nodes %d and %d: ",x,y);
70              scanf("%d",&w);
71              cost [x][y] = cost[y][x] = w;
72          }
73          printf("\n");
74      }
75      printf("\nEnter the source:");
76      scanf("%d", &source);
77      printf("\nEnter the target:");
78      scanf("%d", &target);
79      co = dijsktra(cost,source,target);
80      printf("\nThe Shortest Path: %d",co);
81  }
82
```

<u>How To Use:</u>

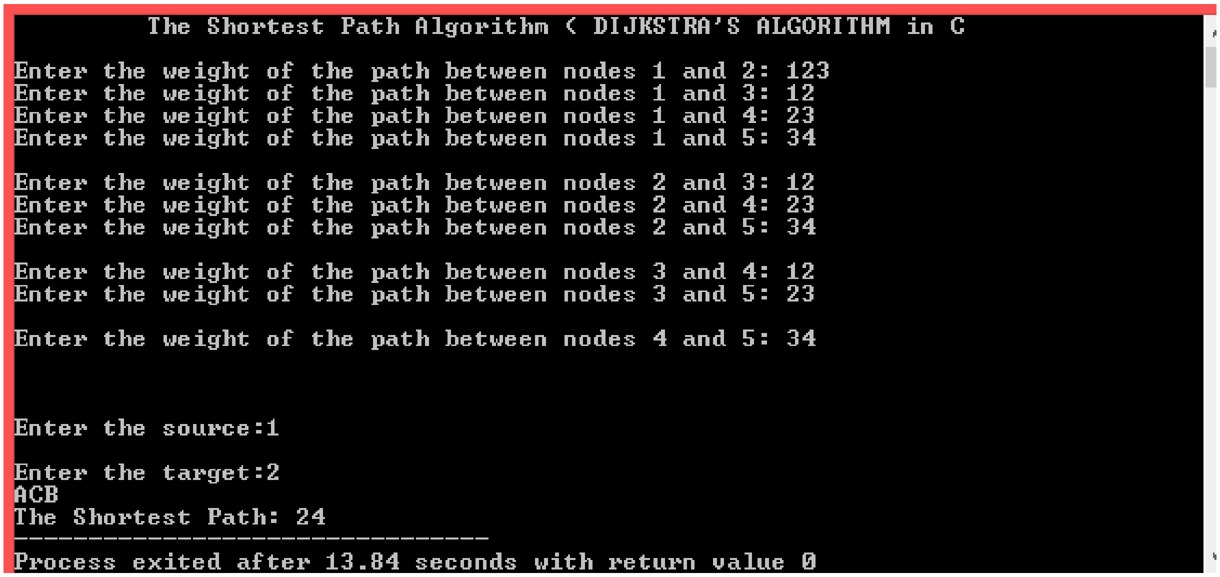
Before you run the program you have to set how many cities (vertices) you have on line 8, make sure to match it to the amount. After you run the program, you will be asked to state the distance between each city (vertices) that stated by the program and after that, state the source and target.

Christine Amelia – 05111740000174

Leonita M. R. S. – 055111740000108

Rimas M. I. – 05111740000185

Input and Output:



```
         The Shortest Path Algorithm ( DIJKSTRA'S ALGORITHM in C

Enter the weight of the path between nodes 1 and 2: 123
Enter the weight of the path between nodes 1 and 3: 12
Enter the weight of the path between nodes 1 and 4: 23
Enter the weight of the path between nodes 1 and 5: 34

Enter the weight of the path between nodes 2 and 3: 12
Enter the weight of the path between nodes 2 and 4: 23
Enter the weight of the path between nodes 2 and 5: 34

Enter the weight of the path between nodes 3 and 4: 12
Enter the weight of the path between nodes 3 and 5: 23

Enter the weight of the path between nodes 4 and 5: 34



Enter the source:1

Enter the target:2
ACB
The Shortest Path: 24
--------------------------------
Process exited after 13.84 seconds with return value 0
```

Analysis:

Based on the picture in Input and Output, we can declare that there are 5 cities (vertices). Each resemble a letter; A, B, C, D, and E. And also there are the declarations of each distance. After being asked which one is the source and target, it will calculate the shortest distance it could find and then state the nodes we should go through to achieve that goal.

Note:

While grade was based on contribution on github, we didn't use it really much since we done this project while gathering together and found that we didn't really need to use that to pull or push the project, so we do hope that our grades it will divide fairly.

Christine Amelia – 05111740000174

Leonita M. R. S. – 055111740000108

Rimas M. I. – 05111740000185

“By the name of Allah (God) Almighty, herewith we pledge and truly declare that we have solved quiz 2 by ourselves, didn’t do any cheating by any means, didn’t do any plagiarism, and didn’t accept anybody’s help by any means. We are going to accept all of the consequences by any means if it has proven that we have been done any cheating and/or plagiarism.”

Surabaya, 30 March 2019

Amelia		
Christine Amelia	Leonita M. R. Sitompul	Rimas Muhammad Irfansyah
NRP 05111740000174	NRP 05111740000108	NRP 05111740000185