Jawahar Education Societys Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

NAME: PRIYUSH BHIMRAO KHOBRAGADE

PRN NO: 211112018

Roll No: 52

SUBJECT: Analysis of Algorithms Lab

Single Source Shortest Path using Greedy Algorithm.

EXPERMINT: 05

PAGE NO.:
DATE.: / / 20

Experiment No-05

• Aim :- Implement a C program for single source shortest path using Greedy Algorithm.

• Objectivies :- To implement and analyze complexity of single source shortest path.

• Out comes :- Students will be able to compute the complexity of single source shortest path.

• Hardware / Software Requised :- 'C' compiler

• Theory

Greedy Approach :- A greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. The principle of optimality state that "In an optimal sequence of decisions or choice, each subsequence must also be optimal."

• Problem Definition :-

Let G(V,E) be the Graph. The problem is to determine the shortest paths from source vertex V_0 to all remaining vertices of G. It is assumed that all the weights are positive. It is also called 'DIJKSTRA'S Algorithm'.

1

Teachers Signature

Single Source Shortest Path using Greedy Algorithm.

PAGE NO.:
DATE.: / / 20

- Algorithm:-

Algorithm SSP (V, ω)

1. current ← ; initialize dist and visit array
2. Initialize distance & visit states of current vertex to 0 & 1 resp.
3. Repeat step 4 to 13 for all vertices.
4. min ← 9999;
5. for i ← 2 to V do
6. if (visit(i) = 0 && dist(i) < min)
7. min ← dist(i)
8. current ← i
9. Change to visit states of the current vertex.
10. for i ← 2 to V do
11. if (ω(current, i) != 0 && visit(i) = 0 && dist(i) > ω(current, i) + dist(current))
12. dist(i) ← ω(current, i) + dist(current);
13. end for

- Analysis

$$T(n) = \sum_{i=1}^{n} 1 \cdot \sum_{j=i}^{n} 1$$

$$= n^2 = O(n^2)$$

- Conclusion:- Thus, it is proved that the complexity of DIJKSTRA's Algorithm is $O(n^2)$.

Teachers Signature

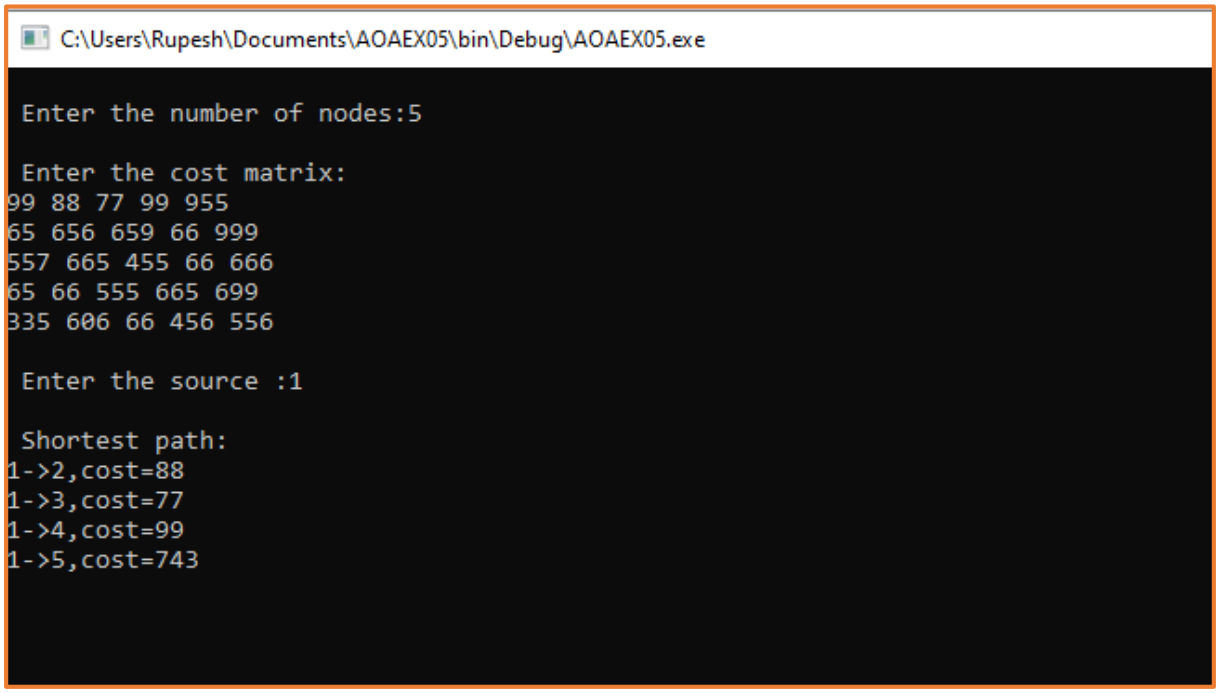
Single Source Shortest Path using Greedy Algorithm.

Input:

```
1 #include<stdio.h>
2  #include<conio.h>
3  #define infinity 999
4
5  void dij(int n, int v, int cost[10][10], int dist[100])
6  {
7    int i,u,count,w,flag[10],min;
8               for(i=1;i<=n;i++)
9               flag[i]=0,dist[i]=cost[v][i];
10               count=2;
11               while(count<=n)
12       {
13               min=9999;
14               for(w=1;w<=n;w++)
15                        if(dist[w]<min && !flag[w])
16                                 min=dist[w],u=w;
17                        flag[u]=1;
18                        count++;
19               for(w=1;w<=n;w++)
20               if((dist[u]+cost[u][w]<dist[w]) && !flag[w])
21               dist[w]=dist[u]+cost[u][w];
22       }
23  }
24  void main()
25  {
26               int n,v,i,j,cost[10][10],dist[10];
27               printf("\n Enter the number of nodes:");
28               scanf("%d",&n);
29               printf("\n Enter the cost matrix:\n");
30                        for(i=1;i<=n;i++)
31               for(j=1;j<=n;j++)
32               {
33                  scanf("%d",&cost[i][j]);
34               if(cost[i][j]==0 )
35               cost[i][j]=infinity;
36               }
37  printf("\n Enter the source :");
38  scanf("%d",&v);
39   dij(n,v,cost,dist);
40  printf("\n Shortest path:\n");
41               for(i=1;i<=n;i++)
42               if(i!=v)
43               printf("%d->%d,cost=%d\n",v,i,dist[i]);
44               getch();
45
46  }
```

Single Source Shortest Path using Greedy Algorithm.

Output:



Conclusion: Thus it is proved that the complexity of Dijkstra's **Algorithm is O(n2).**