SC205 Project

Aksh Patel / 201901005

7thJune 2020

1 Introduction :-

- One day I was thinking about how the pizza delivery boy take different paths to quickly his deliver his orders and also with less cost.
- Here I had made a project to help pizza delivery boy to choose the best path out of all possible paths and also to reduce the travelling cost!!!
- I will use some $DISCRETE\ MATHEMATICS$ topics like -

1. **Graph Theory**
   a). Adjacency Matrix
   b). Hamiltonian Circuit
   c). Vertices
   d). Paths
2. **Sets**
   a). To store nodes
3. **Matrices**
   a). To store Graph in Matrix form.
   b). To store all the related calculation in Matrix form.
4. **Logic Conditions**
   a). To decide which path to take to get maximum benefit.
5. **Time Complexity** (θ, Ω, O notations)
   a). To measure the running time of both the algorithms.
   1. Dijkstra Algorithm.
   2. Travelling Salesman Problem.

Note : I had also done the software implementation part of this project in C.

2 Problem Statement :-

As shown in Fig : 1 a pizza delivery boy starts from Restaurant (R) and he want to go to places D , E and F for delivering pizza.

So, You have to find out which path he takes to minimize his cost and also the minimum cost.

The cost of respective paths are mention on paths in Fig : 1 .

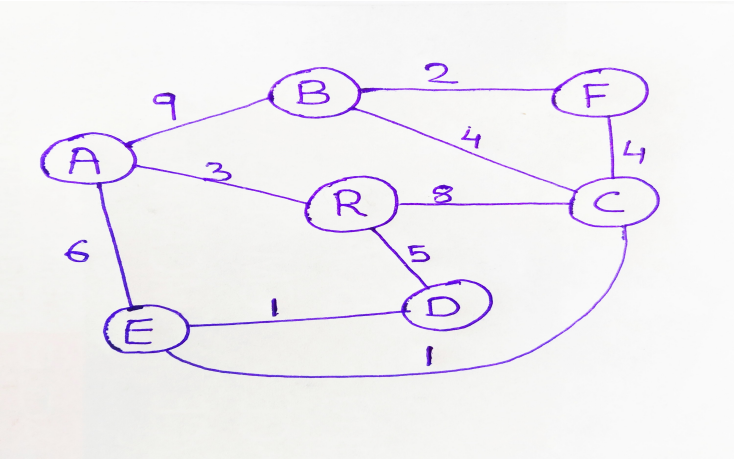

Figure 1: Finding best path with Less Cost.

Adjacency Matrix : It stores the whole graph in matrix form . It specifies the distance between two nodes.

Adjacency Matrix	R	A	B	C	D	E	F
R	0	3	0	8	5	0	0
A	3	0	9	0	0	6	0
B	0	9	0	4	0	0	2
C	8	0	4	0	0	1	4
D	5	0	0	0	0	1	0
E	0	6	0	1	1	0	0
F	0	0	2	4	0	0	0

Table 1: Adjacency Matrix

3 Algorithm or Procedure :-

3.1 Finding shortest Path Distance OR Minimum cost :

– First I will find shortest paths from nodes R , D , E and F to all the remaining nodes by Dijkstra Algorithm.

3.1.1 Procedure to find shortest path distance :

1. Make a Matrix of size $(n * n)$ where n is the total no of nodes in graph model.
2. First column in matrix denotes selected vertex.
3. Remaining columns shows the distance to corresponding vertex.
4. So, first we have to select the vertex from which we want to find shortest distances to all the vertices.
5. If any vertex is directly attached to the selected vertex then we put their distance in matrix otherwise we put infinity.
6. Now we have first row ready with us. So we go through from 2^{nd} column to last column in 1^{st} and find the smallest value of 1^{st} and select that vertex in the 2^{nd} row.
7. If any vertex is directly attached to the selected vertex then we check if the distance from the selected vertex is smaller than previously modified than we change the previous distance to the distance between them in that column otherwise we left it unchanged.
8. We repeat this procedure untill all the vertices are selected.
9. Last row that we get in this matrix denotes the smallest distance of that vertex to the first selected vertex.
10. During this procedure we have to make a traceback matrix to get the best path out of many possible paths. [2]

3.1.2 Calculation of Matrices of R , D , E , F :

1. From node R

Selected Vertex	A	B	C	D	E	F
R	3	∞	8	5	∞	∞
A	3	12	8	5	9	∞
D	3	12	8	5	6	∞
E	3	12	7	5	6	∞
C	3	11	7	5	6	11
B	3	11	7	5	6	11
F	3	11	7	5	6	11

Table 2: Calculation of shortest distance from R.

- First we select vertex R and do procedure as mention in 3.1.1.
- Then we select the vertex with smallest distance, in this case it is A.
- Then we iterate from 2^{nd} to last column and if the distance from A to that vertex is smaller than previous value change the value to distance between them else keep as it is.
- Now repeat this till all vertices are selected.

Shortest Distance from vertex R to	A	B	C	D	E	F
	3	11	7	5	6	11

2. From node D

Selected Vertex	R	A	B	C	E	F
D	5	∞	∞	∞	1	∞
E	5	7	∞	2	1	∞
C	5	7	6	2	1	6
R	5	7	6	2	1	6
B	5	7	6	2	1	6
F	5	7	6	2	1	6
A	5	7	6	2	1	6

Table 3: Calculation of shortest distance from D.

Shortest Distance from vertex D to	R	A	B	C	E	F
	5	7	6	2	1	6

Table 4: Results of D

3. From node E

Selected Vertex	R	A	B	C	D	F
E	∞	6	∞	1	1	∞
C	9	6	5	1	1	5
D	6	6	5	1	1	5
B	6	6	5	1	1	5
F	6	6	5	1	1	5
R	6	6	5	1	1	5
A	6	6	5	1	1	5

Table 5: Calculation of shortest distance from E.

Shortest Distance from vertex E to	R	A	B	C	D	F
	6	6	5	1	1	5

Table 6: Results of E

4. From node F

Selected Vertex	R	A	B	C	D	E
F	∞	∞	2	4	∞	∞
B	∞	11	2	4	∞	∞
C	12	11	2	4	∞	5
E	12	11	2	4	6	5
D	11	11	2	4	6	5
R	11	11	2	4	6	5
A	11	11	2	4	6	5

Table 7: Calculation of shortest distance from F.

Shortest Distance from vertex F to	R	A	B	C	D	E
	11	11	2	4	6	5

Table 8: Results of F

- Now from the above calculation of 4 vertex we will make a simplified graph which contains only these 4 vertices and paths between 4 these vertices only.
- And the simplified graph is shown below in $Fig : 2$.

[3]

3.2 Finding best Hamiltonian Circuit from all possible paths:

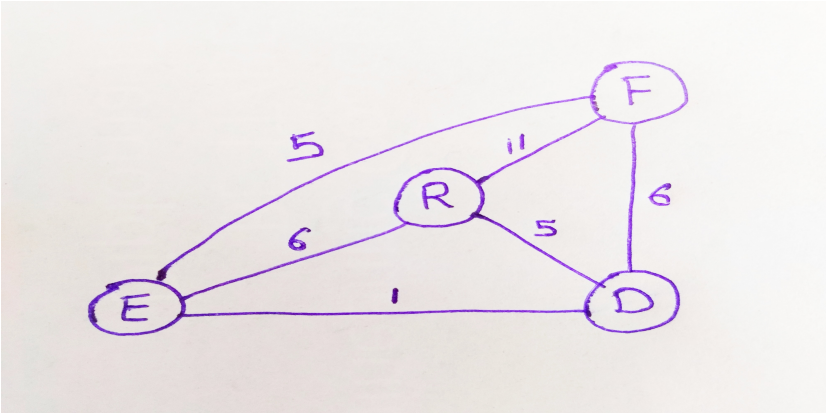

Figure 2: Simplified Graph containing less vertices than original Graph.

Hamiltonian Circuit : A simple circuit in a graph G that passes through every vertex exactly once is called a *Hamiltonian Circuit*.

In this section we have to find best Hamiltonian circuit with minimum distance.

3.2.1 Problem :

–Pizza Delivery boy has to start at Restaurant (R) and he want to visit all the nodes exactly once and return to the Restaurant(R).

3.2.2 Recursive Procedure :

1. As shown in recursive tree in $Fig : 3$, we start at the source vertex R and from R we have 3 ways to go.
2. And from that 3 ways we have 6 ways and we have completed till levet 2.
3. Now at level 3 we have 6 ways.
4. From that 6 ways at level 3 we have to find distance of these 6 vertex with source vertex R.
5. Now we have to go reverse from bottom to top of the tree and keep adding distances between nodes and if at some vertex we get two values then we have to select the minimum distance and go above.
6. In this way at root we have 3 values of distances so we have to compare theses distances and select the minimum distance.

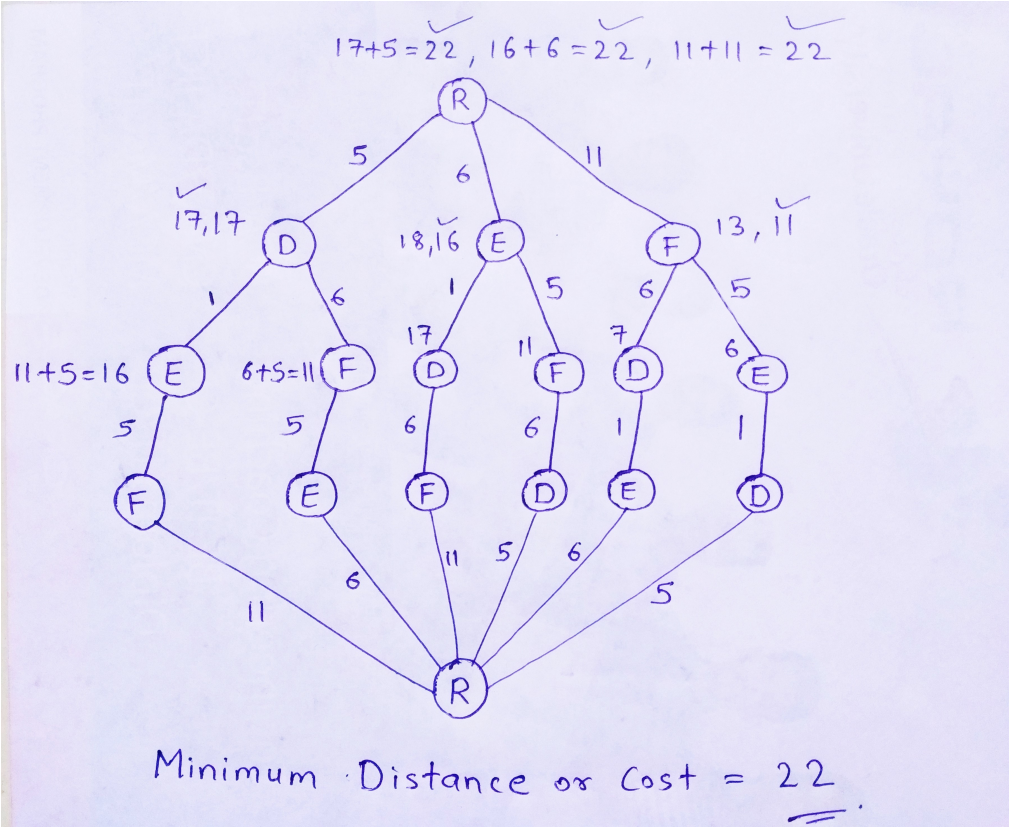

Figure 3: Recursion Tree:

–Recursive formula for the above tree is -

$$g(i, S) = min_{k \epsilon S}\{C_{ik} + g(k, S - k)\}$$

where $g(i, S)$ is the recursive function,S is the set of all vertices,k is the selected vertex and C_{ik} is the distance between i and k.

Eg: S={D,E,F}
k is D.
$S - \{k\} = \{E, F\}$
C_{RD} is the distance between R and D.
$g(R, S) = min_{D \epsilon S}\{C_{RD} + g(D, S - D)\}$

- So the minimum distance that pizza delivery boy must travel is 22 units.
- And one of the best path is $R - D - E - F - R$. [1]

3.3 Finding Time Complexity

1. **For Dijkstra Algorithm** (*Section*3.1)**:**

   - In this algorithm we have to select every node one by one and for each node we have to check for adjacent nodes distances.
   - For, worst case we have $(n-1)$ adjacent nodes for every n nodes.
   - Hence, wort case Time Complexity is $O(n*(n-1)) = O(n^2)$.
   - For best case we have 1 adjacent node for every n nodes.
   - Hence Best case TIme Complexity is $\theta(n*1) = \theta(n)$.

2. **For Finding Best Hamiltonian Circuit** (*Section*3.2)**:**
   –Recursive formula for the above tree is -

$$g(i,S) = min_{k \epsilon S}\{C_{ik} + g(k, S-k)\}$$

   - This algorithm recursively finds the shortest tour starting from each neighbor and returns the minimum of those.
   - If you do all this with dynamic programming you can calculate the amount of work you're doing like so:
   - There are n possible start vertices and 2^n possible subgraphs.
   - So this function will be called on at most $n.2^n$ distinct arguments.
   - Each call performs at most $O(n)$ work (there are at most n neighbors).
   - Hence the total work you're doing is $O(n^2 2^n)$. [4]

4 Conclusion :

- So by this solution we can find the shortest path and also the shortest distance or say minimum cost.
- This process is little bit same as the logic behind the google maps.
- If we modify certain things then we can do mny things out of it.
- We can also do some research and make another algorithm for this problem and make it even more efficient.

5 Commercialization of this Product :

This type of Project has high commercial requirement.And the proof of this is that we are using Google Maps in our everyday life to go to our destination with faster and with minimum cost or the minimum distance.

–My Project is little different from Google Maps and the differences are :

1. Google maps only calculates distance, time and shortest path.
2. On the other side my project calculates minimum distance and the best possible path.
3. But the major difference from Google Maps is that my project calculates shortest path from one node to other and also it finds the shortest path and minimum distance if we want to visit v nodes out of n nodes like a pizza delivery boy does or for any courier service.

-It has many advantages like :

- By using this technology we can save our fuel and hence we preserve our natural resources from being destroy rapidly.
- If less fuel is used then we can save our money.
- We can also decrese pollution by this way.
- Finally the most important thing that we save by using this is **Time** which is very precious than anything.

We can make an app and upload it on Play store to do our own startup and **earn money** out of it!!!

References

[1] Saheb Ghosh. Travelling salesman problem , with c program example. https://sahebg.github.io/computersceince/travelling-salesman-problem-c-program/, 02-08-2017.

[2] N. Jasika, N. Alispahic, A. Elma, K. Ilvana, L. Elma, and N. Nosovic. Dijkstra's shortest path algorithm serial and parallel execution performance analysis. In *2012 Proceedings of the 35th International Convention MIPRO*, pages 1811–1815, 2012.

[3] Neeraj Mishra. Dijkstra's algorithm in c. https://urlzs.com/W2B6i.

[4] Sebastian Oberhoff. Analysis of time complexity of travelling salesman problem. https://cs.stackexchange.com/questions/90149/analysis-of-time-complexity-of-travelling-salesman-problem, 03-04-2018.

Thank You