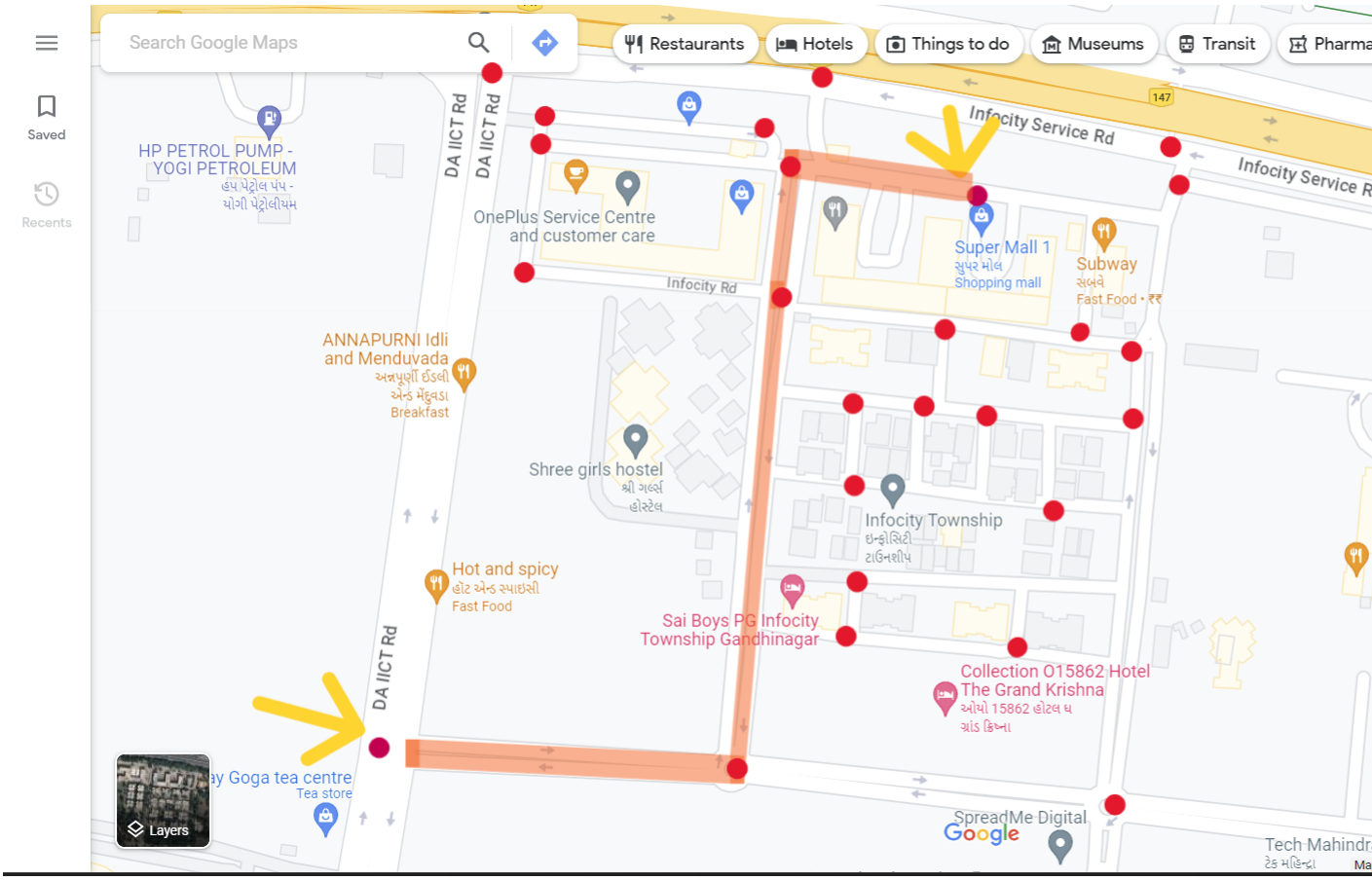
Use of Discrete Mathematics in Google Maps

May 23, 2023

How maps works ?

1 Google Maps working

Have you ever wondered how google maps or any other map based AI work ? To think of it we shall see some major aspects and mathematics related to this first.
The main use is of a algorithm known as Dijkstra's algorithm and a A* algorithm, both are used for defining location and giving the user the shortest path-length to reach the required location.
Talking about the first algorithm , this algorithm at first was used to get shortest paths between the nodes. On upgrading it a bit it maintained one of the given node as main point or rather a source and finds the shortest path related to other nodes.
The second algorithm is more focused on the final nodes to reach the required destination more efficiently and quickly.The google maps use both of the algorithms to improvise and make a good use of them for the user.



(1)

Figure 1:

2 Using map theory and finding distance

Consider the following map. All the red dots represent the nodes. Google map can also be explained via nodes. If we consider these nodes, suppose our location is the node with arrow below and we want to reach the location a super mall. Then what could be the best way to go.

There are many nodes we can select to make our way to the location but the best is needed. The google maps using the Dijkstra's Algorithm and A* algorithm helps to serve the required needs. It first check the nodes , calculates the minimum number of nodes required or in a way shortest distance between the nodes to reach a particular coordinates.

It checks in our way if there is any node which satisfy the condition of being the shortest node and if it is then replaces it with the new node which is said to be the shortest path of node.

Taking the following example the best path with nodes of shortest distance is chosen. We have 5 nodes till we reach the required location. By using the graph theory , study of points and lines, the distance between the nodes is calculated first between node 1 and 2 , then between 2 and 3 and similarly till last .

After that we compare the total distance taken by all nodes, and chose the one with the shortest distance. Of course the the calculation is a bit more work but algorithm makes it more simple to do. The coding enables us to make this more meaningful and fast .

The code we used for finding the shortest distance between the node is given as :

The given code enables us to get the Shortest distance between nodes. The loop will check if there is any shorter distance that satisfies the given conditions and if there is than replaced the shortest path with the previous one. The code here works in the following ways :

First the code takes the input from the user, then a loops runs and input of the length on the nodes is taken. After the loops we set the final and initial node. Now the loop compares the node's distance n times('n' the number of nodes entered) with other node's distance.

There the three certain for that to compare . After comparing the three nodes and the distance the final distance or the shortest distance is shown in the output.And thus we get the nodes with the shortest distance after comparing with other nodes.

#include iostream

File Edit View

```
const int MAX_NODES = 10;

int main() {
    int nodes, start, final;
    int distance[MAX_NODES][MAX_NODES];

    cout << "Enter the number of nodes: ";
    cin >> nodes;

    for (int b = 0; b < nodes; b++) {
        for (int c = b+1; c < nodes; c++) {
            cout << "Enter the distance between node " << b+1 << " and node " << c+1 << " : ";
            cin >> distance[b][c];
            distance[c][b] = distance[b][c];
        }
    }

    cout << "Enter the starting node: ";
    cin >> start;
    cout << "Enter the final node: ";
    cin >> final;

    int d_distance[MAX_NODES], finaldistance;

    for (int d = 0; d < nodes; d++) {
        d_distance[d] = distance[start-1][d];
    }

    finaldistance = d_distance[final-1];

    // find the shortest distance from start to final node
    for (int i = 0; i < nodes-2; i++)
    {
        for (int j = 0; j < nodes; j++)
        {
            if (j != start-1 && j != final-1)
            {
                int new_distance = d_distance[j] + distance[j][final-1];
                if (new_distance < finaldistance) {
                    finaldistance = new_distance;
                }
            }
        }
        for (int k = 0; k < nodes; k++)
        {
            if (k != start-1 && k != final-1)
            {
                int new_distance = d_distance[k] + distance[k][final-1];
                if (new_distance < d_distance[final-1])
                {
                    d_distance[final-1] = new_distance;
                }
            }
        }
    }

    cout << "The shortest distance between " << start << " and " << final << " is: " << finaldista

    return 0;
}
```

The main conclusion from above algorithm and the code is for a user who wants to reach the location with shortest and best possible way.Thus using the knowledge of graph theory that explain the nodes relation and algorithms like Dijkstra's and A* we made a code to analyse the nodes form starting/initial point to final point and then to find the shortest path distance relating to the initial node.

2.1 Commercial aspects related to maps service

Maps are in the very need of daily life a person. Almost in the total population approximately 67 percentage people use maps per hour. The map services serves as the major component of the commercial aspect of google. Taking about our hand the main thing to improvise the map services is to make a website than further improving it. The commercial aspect is mainly based in user friendly appearance and use. The more better it is user friendly the more users we get.

So if we were to expand this map services we would make a software that is more user desired and defined.

Furthermore we can sponsor the locals to improve at local level , add more sponsors like the local shops, their ratings and information. If we tend to upgrade it more we could let the users apply for sponsoring their business and ideas , showing it in maps , making it available to people who are travelling the area or are passers.

This can be of great help to local and also to increase the popularity of the map services among people. Further more adding more people to the team , making a application to be more user convenient. Making it more better for the locals to use will lead to good rise in commercial aspect. If these to done than there can be adding of more business ideas and sponsors apart from locals, making it more professional suitable not only for locals but also for business professionals.

Thus improving would be the major task to improve in commercial aspect. Improving the precision of maps , and details in it would be a necessary part of it. The real world situation can be different compared to virtual so in order to be more popular , the more specific details should be taken care of , the changes in maps and all. It must be updated. User should be able to use the maps services more efficiently and conveniently.

3 Conclusion

We saw form how the map services work and how to deal with them with some examples and theories , from the mathematical approach to the commercial. On conclusion the major part was not only to find the shortest distance between the nodes but also to establish and improve map services depending in the user guidance and approach towards making it more easy to use and allowing people to sponsor their business.

Improving it more makes each step toward the maps services and it capital nature. We talked about how the google maps work , how they use different algorithms and apply in their maps efficiently, used examples , and mathematical approach to explain the nodes relation, talked about codes , working and practical point of view , we also talked about the commercial view and its working.(Learning more and more about the graph theory and maps)