Emergency Vehicle Dispatching System

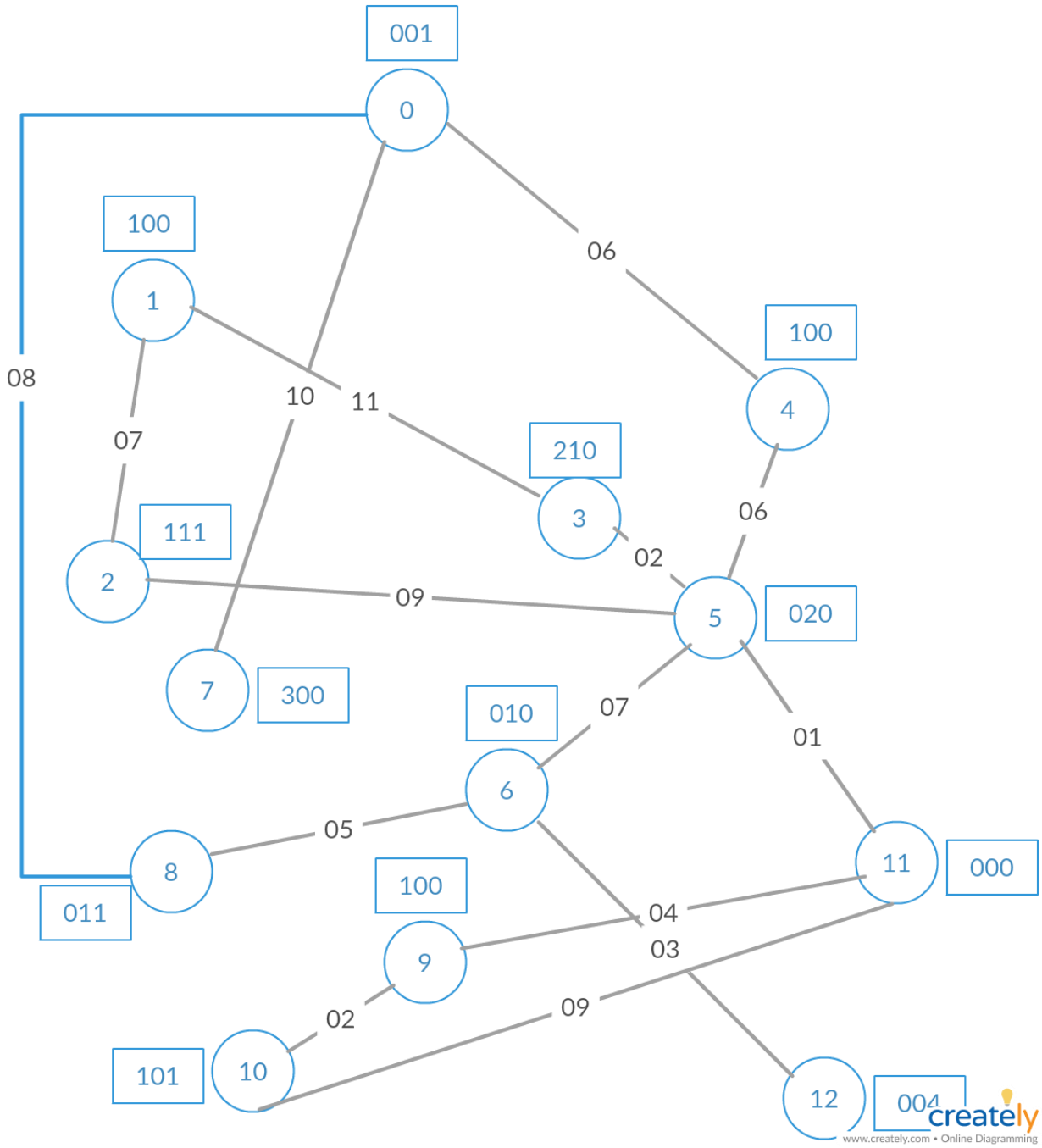
By : Uday Kiran Dora Sevana,

Girish Kumar Reddy Nagella,

Mani Sai Srinivas Kandukuri,

Sai Mohith Reddy Chagamreddy.

Emergency Vehicle Dispatch System.



The above is the graph for the Emergency vehicle dispatch system.

Any references to the nodes or distances will be referring to the image above.

Assumptions:

1. The vehicle once dispatched from a zip code will not return to it.
2. The amount of time taken is dependent on distance only. So the closest vehicle will always be the first to reach the destination, hence dispatched first.
3. Vehicle Id's are not in order. There is no need that the vehicle with id no.1 will be dispatched before the vehicle with id no.2.
4. A zipcode can have any number of neighbors.
5. The unit of distance is in miles. The time for the vehicle is in miles per minutes.
6. There is no constraint on the number of vehicles to be present at the nodes. It can be of any value or none at all. Our chosen values are The number of fire truck 10, police 5 and ambulance 9.

Brief Explaination:

The problem statement: To request an emergency vehicle from the available 3 types, from any zipcode and to fetch the closest available vehicle.

The zipcodes: The base idea to tackle the problem is to find the closest nodes and check if the requested emergency vehicle is available at the selected node, if not then move to the 2nd closest node and so on until the requested vehicle id dispatched.

Nodes to zipcodes

0.64110 **1.**64130 **2.**64127 **3.**64134 **4.**64030 **5.**64149 **6.**64155

7.64064 **8.**64143 **9.**64210 **10.**65311 **11.**61255 **12.**61232

The Vehicle: The number of the available emergency vehicle are limited, this might result in different nodes dispatching a different vehicle in consecutive requests. The vehicle once dispatched from a node is no more available at that node. Hence decrementing the number of emergency vehicles available.

The Idea: The algorithm used to solve the problem is Dijkstra's algorithm. The base idea of Dijkstra is to find the shortest path to every other node present in the graph. Using this idea to dispatch the vehicles, a list of availability is assigned to every node (from hereon the zip codes will be referred to as nodes). With this list being fixed the node closest to the requesting node with a vehicle available will dispatch it.

For example

Let the requesting source zip code be 64110—node 0
The distances of every node from node:0 are //found using Dijkstra Node 0: 0 (zero being itself)
Node 1: 25 **Node 2:** 21 **Node 3:** 24 **Node 4:** 6 **Node 5:** 12 **Node 6:** 13
Node 7: 10 **Node 8:** 8 **Node 9:** 17 **Node 10:** 19 **Node 11:** 13 **Node 12:** 16

If a Police vehicle is requested even though node 4 is the closest, it does not have the emergency vehicle required. Hence the requested vehicle is dispatched from node 5.

This can be done by iterating through the dorted distances array and then checking for the availability of the requested vehicle for every node encountered in the sorted array. If the vehicle is not available in that particular array the next node is checked for the availability.

0	4	8	7	5	6	11	12	9	10	2	3	1
0	6	8	10	12	13	13	16	17	19	21	24	25
0	0	0	0	2								

It can be seen above that all the nodes before 12 have no police vehicles avaliable but node 5 has 2 so one is dispatched.

```
enter Source zipcode of avaliable
1.64110 2.64130 3.64127 4.64134 5.64030 6.64149 7.64155
8.64064 9.64143 10.64210 11.65311 12.61255 13.61232
64110
enter type fire:1 Police:2 Ambulance:3
2
Vehicle P46 Available from '5'and is on it's way to destinationwill arrive in 12mins
```

The Above is the command line output for the program.

The time is indirectly the distance of the vehicle from requesting zipcode

Efficiency of Algorithm

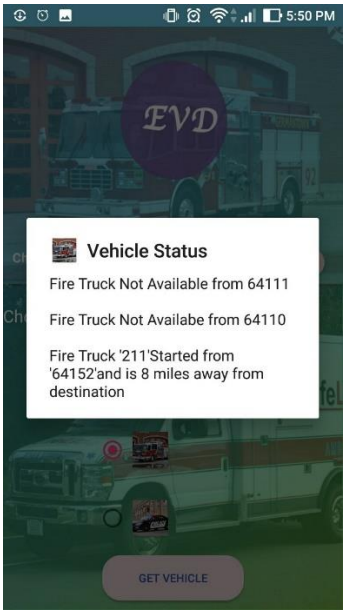
The effeciency of the Dijkstra's algorithm is $O(V^2)$ where V is the number of edges.

The effeciency of the algorithm with the added functions will also be close to $O(V^2+n^2)$ as there are extra number fo iterations. But the number of iterations n is also based on number of vertices V so it is $\sim O(V^2)$

It can be improved using priority queue, when implemented with binary heap and hash map. The big O is reduced to O(E log V)

Android Application

Below is the android application designed for the project.



References:

Geeks for geeks