**A PATH PLANNING ALGORITHM FOR BLIND PEOPLE**

## Submitted by

## AKSHAT JAIN (2K19/IT/010)

## AMARDEEP SINGH (2K19/IT/015)

Under the Guidance of Ms. Swati Sharda



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of

Engineering) Bawana Road,

Delhi- 110042

DECEMBER, 2020

### CANDIDATE’S DECLARATION

I hereby certify that the work which is being presented in the project entitled

**A Path Planning Algorithm For Blind People"** submitted in the **Department of** Information Technology of the Delhi Technological University, Delhiis an authentic record of our own work carried out for the requirement of the Data Structures course IT-205 in the 3rd Semester 2020-2021.

Place: Delhi

Date: 25-11-2020

Akshat Jain

Amardeep Singh

**CERTIFICATE**

This is certified that the above statament made by the candldates is correct to the best of my knowledge.

Place: Delhi

Date: 25-11-2020

**Swati Sharda**

**SUPERVISOR**

# Acknowledgement

I am heartily thankful to my guide, Ms. Swati Sharda, whose encouragement, guidance and support enabled me to develop an understanding of the subject and made it possible for me to complete this project. I offer my regards to all the respected professors that have taught me since their contribution towards my knowledge has been immense enabling me to work on this project.

-Akshat Jain, Amardeep Singh

ABSTRACT

Navigation is a fundamental requirement of every individual. People have their own well-defined criteria for choosing a path when moving from one place to another. For a school student or an office employee who is getting late for the school or the office, the shortest path may hold the utmost importance; a fitness freak or a health conscious person may choose a longer route; an elderly would most likely choose a well-lighted path with least hurdles and so on. But visually impaired people are not privileged to make their own choices when navigating from one place to another. They usually try to memorize routes for every environment they visit which besides been a tedious task, can also become problematic in the case of a change in the architecture of the building, an under- construction road or the presence of a dynamic or static obstacle that could vary from his previous experience. Visually impaired people cannot even try out a new route as they lack the information about the surroundings and feel prone to various kinds of risks involved. Similarly, they also require constant assistance for navigating through the supermarket and choosing their desired product.

CONTENTS

* Certificate
* Acknowledgement
* Abstract

1. Introduction to Path Planning Algorithm
2. Implementation of Algorithm
   1. Theory
   2. Flow Chart
   3. Code
   4. Output
3. Dijkstra Vs Optipath
4. Conclusion
5. **Introduction to Path Planning Algorithm:**

The proposed algorithm aims at guiding the visually impaired people through the various aisles in the supermarket from their current position to their choice of destination so that they could carry out an assistance free and independent shopping in the supermarkets. The algorithm assumes a single floor supermarket layout with shelves on both sides of the path. The proposed algorithm computes an optimal path from the user’s current position to chosen destination. If an obstacle is identified during the navigation through the suggested path, the algorithm reroutes the user to his destination through a new optimal path.

The input to the path finding algorithm consists of a graph corresponding to the supermarket layout, user’s current position, user’s chosen destination and the list of obstacles. The output is the obstacle- free optimal path from user’s current position to his chosen destination in the supermarket.

1. **Implementaion of Algorithm:**

**A.THEORY:**

**GRAPH CONSTRUCTION:**

In order to implement the proposed algorithm for a supermarket, it is required to get the supermarket design layout and convert it into a corresponding undirected weighted graph. The supermarket graph construction follows the following rules: a) Every section in the supermarket is represented by a node. b) Certain weight is assigned to each edge in the graph based on the distance between the two nodes. c) A single node is used to represent the sections on opposite sides of the path.

**MATRIX CONSTRUCTION:**

**Weight matrix**

* W[i,j]=the distance from node i to j.
* =10000 if (i,j) is not an edge in the graph or i=j

**Normalised Weight matrix**

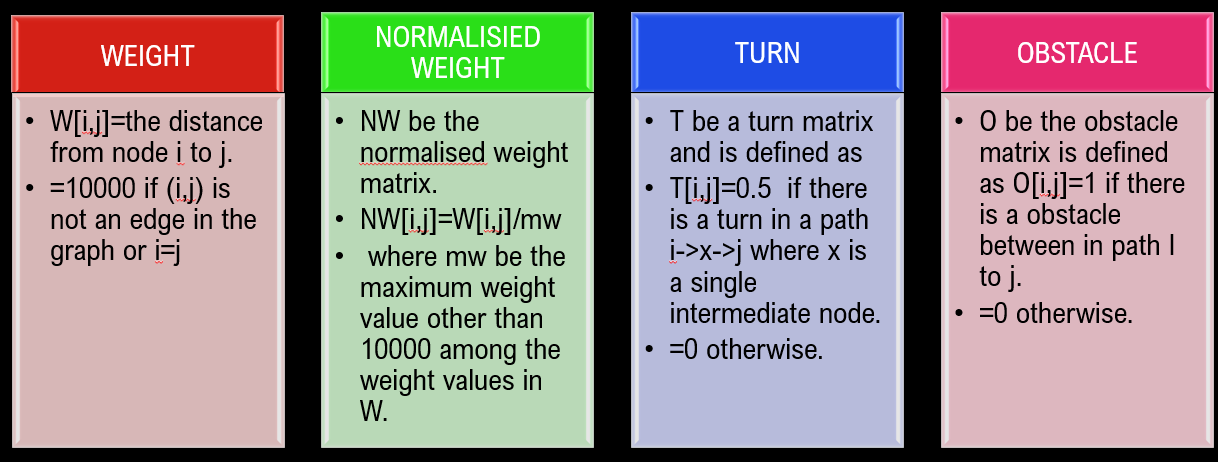
* NW be the normalised weight matrix.
* NW[i,j]=W[i,j]/mw
* where mw be the maximum weight value other than 10000 among the weight values in W.

**Turn Matrix:**

* T be a turn matrix and is defined as
* T[i,j]=0.5 if there is a turn in a path i->x->j where x is a single intermediate node.
* =0 otherwise.

**Obstacle Matrix:**

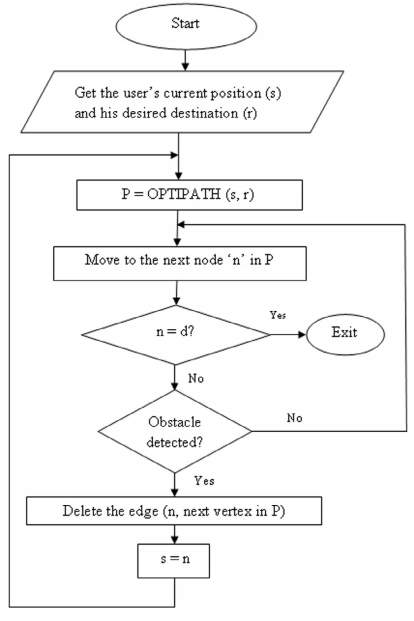
* O be the obstacle matrix is defined as O[i,j]=1 if there is a obstacle between in path I to j.
* =0 otherwise.



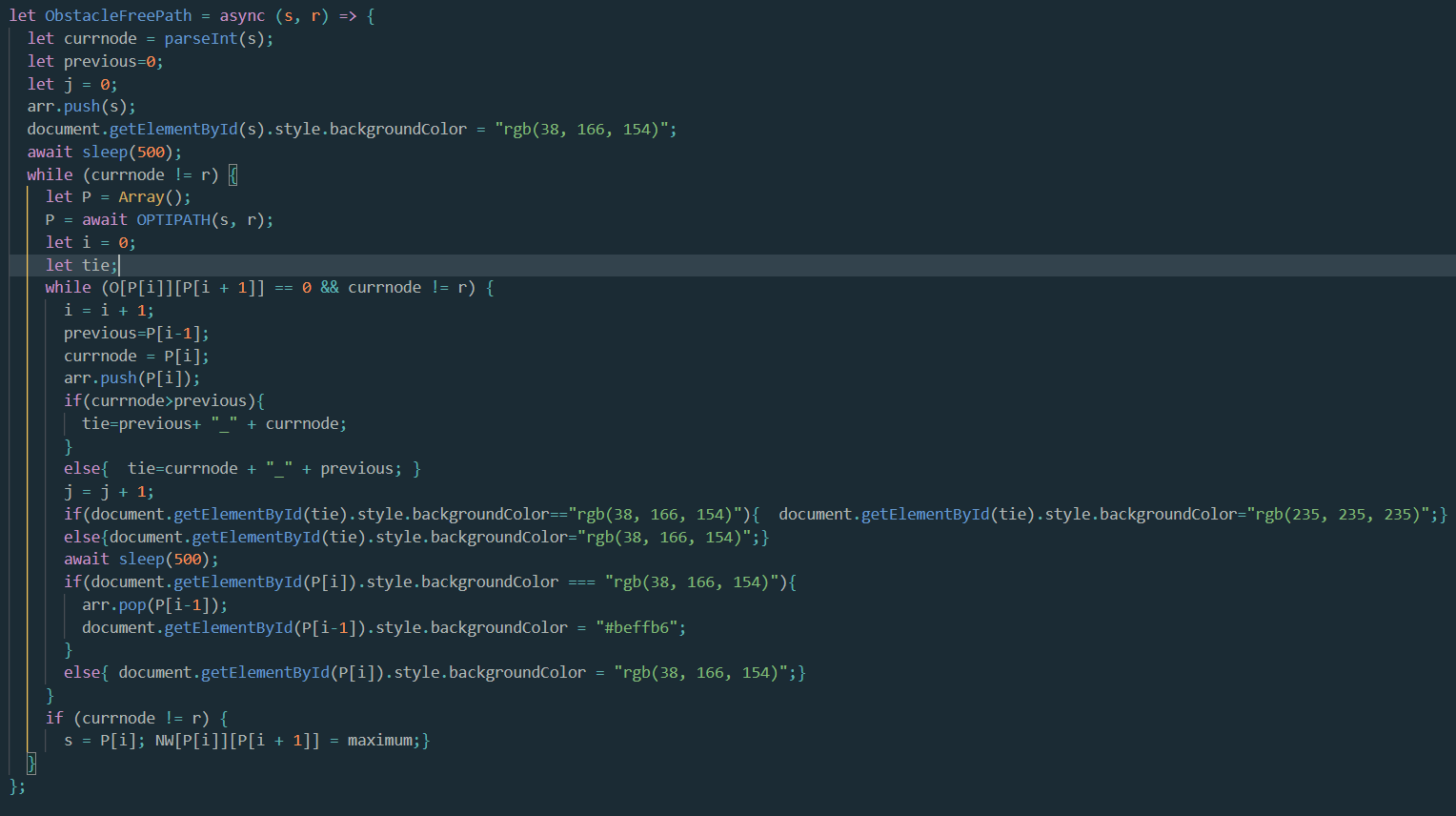
**Proposed methodology:**

The various steps in the proposed methodology are 1. Get the supermarket design layout. 2. Convert the supermarket layout into an undirected weighted graph. 3. Normalize the weight matrix (W) by dividing it by the maximum of all the edge weights so that all weight values are within the range [0,1]. 4. Assign a weight of 0.5 to each turn, since 0.5 is the median of the range [0,1]. 5. Optimum distance = distance travelled + number of turns ⁄ 0.5. 6. Call OPTIPATH () which is a greedy algorithm to find the optimal path from source to destination. 7. Traverse the path, checking for obstacles. 8. If Obstacle is found delete the corresponding edge from the graph and call OPTIPATH () with current node as the new source. 9. Continue until the destination is reached.

**B. FLOW CHART:**

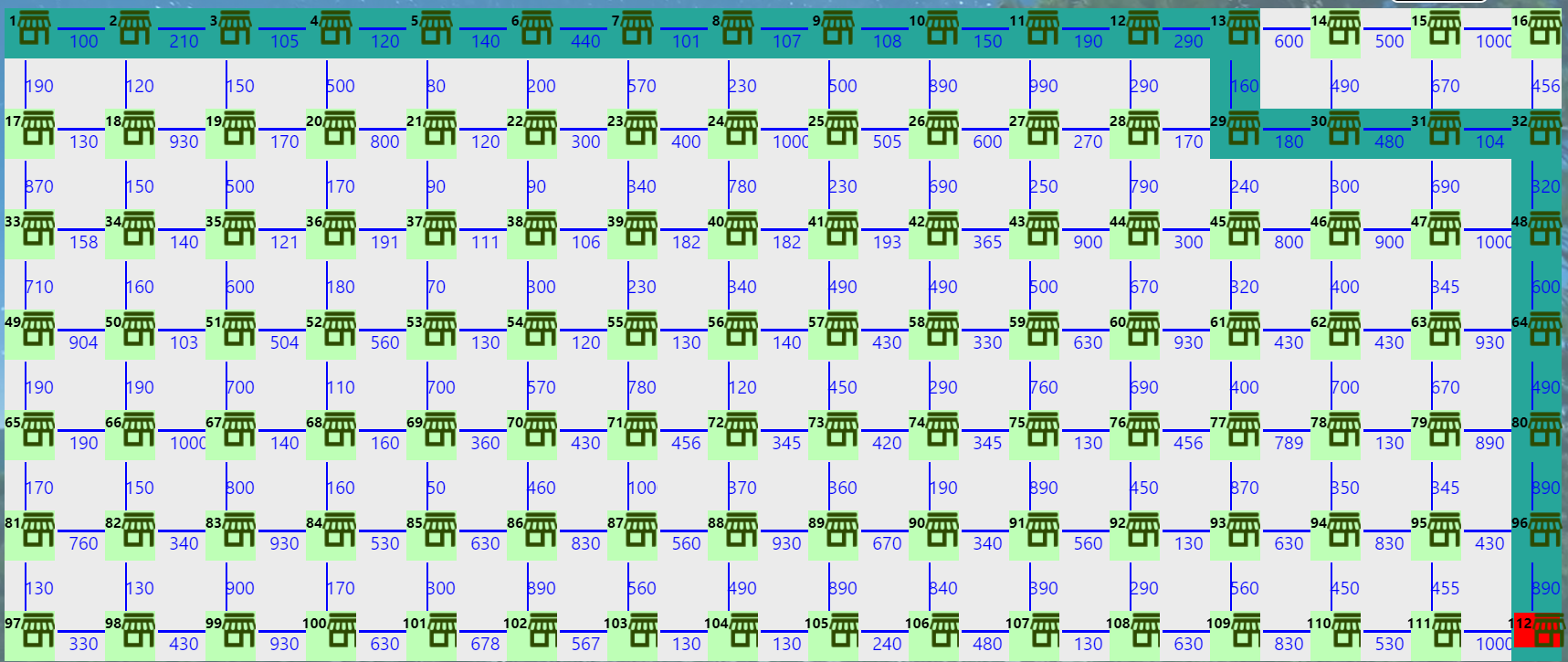


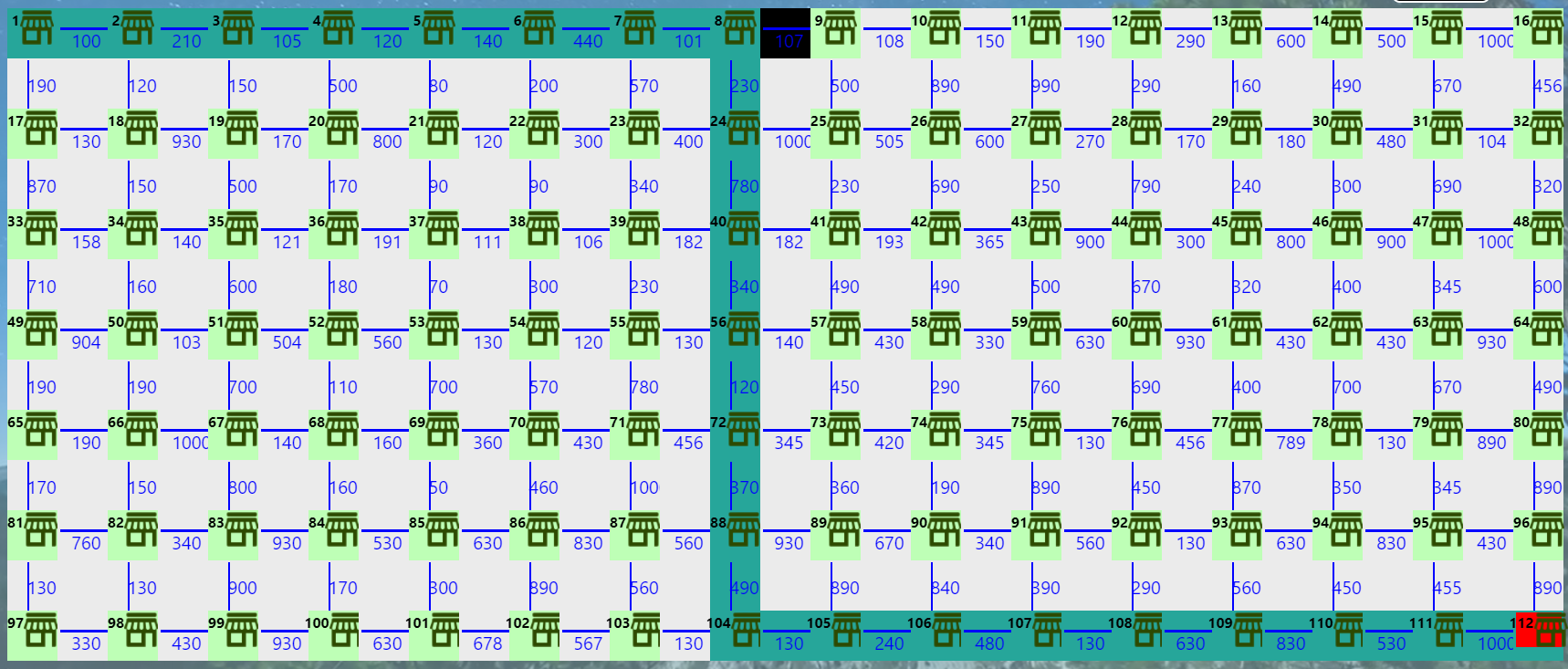
**C. CODE**

****

****

**D. OUTPUT**





1. **Dijkstra Vs Optipath**

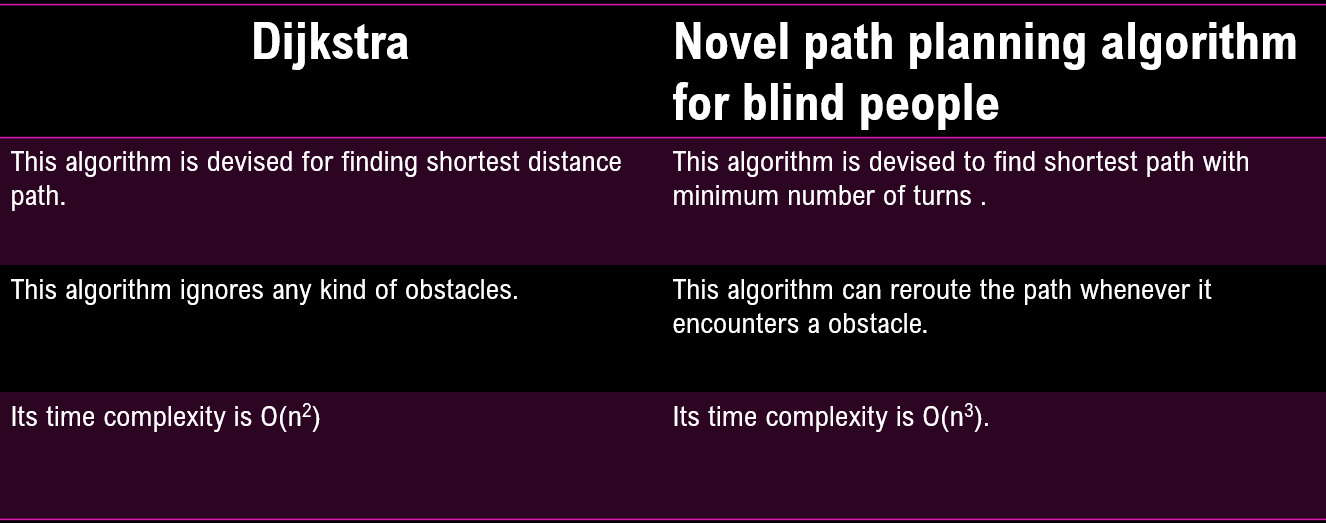
**Dijkstra**

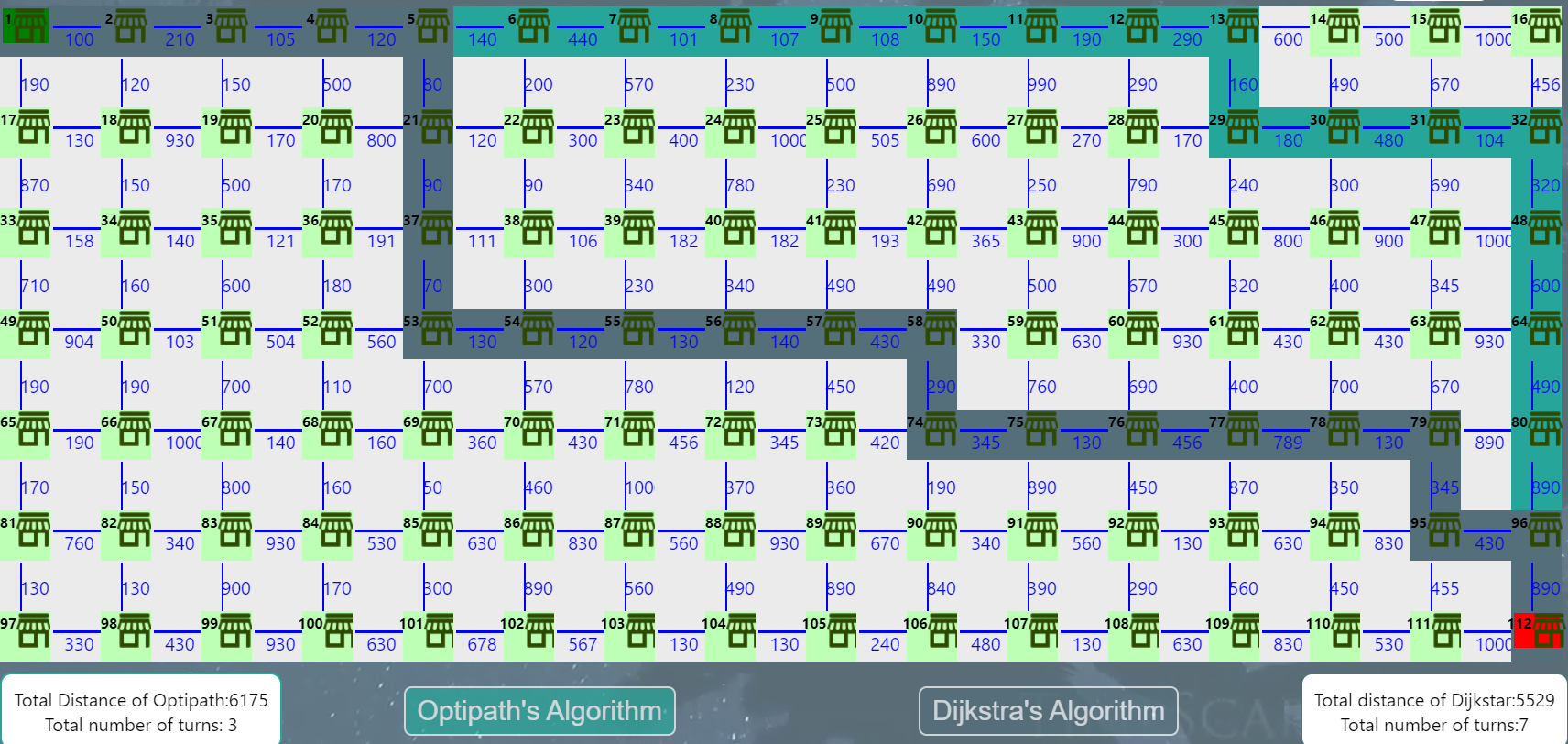
Dijkstra’s algorithm is very similar to [Prim’s algorithm for minimum spanning tree](https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/). Like Prim’s MST, we generate a SPT (shortest path tree) with given source as root. We maintain two sets, one set contains vertices included in shortest path tree, other set includes vertices not yet included in shortest path tree. At every step of the algorithm, we find a vertex which is in the other set (set of not yet included) and has a minimum distance from the source.

Below are the detailed steps used in Dijkstra’s algorithm to find the shortest path from a single source vertex to all other vertices in the given graph.  
Algorithm  
**1)** Create a set sptSet (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.  
**2)** Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.  
**3)** While sptSet doesn’t include all vertices  
….**a)** Pick a vertex u which is not there in sptSet and has minimum distance value.  
….**b)** Include u to sptSet.  
….**c)** Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.

The proposed algorithm has been implemented inJava script. Various experiments were conductedby taking a sample supermarket scenario. OPTIPATH() aims to plan a safest, shortest and minimum turn path for the visually impaired user and re-route the user to his destination if he gets stuck in any section of the supermarket. Suppose a visually impaired customer ‘X’ enters into the supermarket and wishes to purchase some milk and bread from the dairy section in the supermarket. Then the user ‘X’ provides the destination as input i.e. the dairy section, which is one of the inputs to the proposed algorithm. The other input is the user’s current position in the supermarket that is retrieved by the available assistive system. The proposed algorithm, OPTIPATH() finds the optimal route for the user such that the suggested route is short and has less number of turns. The OPTIPATH() also re-routes the user from

his current position to his choice of destination if an obstacle is detected in the suggested path.





1. **CONCLUSION:**

A novel path planning algorithm for visually impaired people is proposed in this paper. The proposed algorithm can be incorporated into existing navigational aids like an electronic white cane used by the visually impaired people. The major feature of the algorithm is the minimization of a number of turns which is one of the preferences of visually impaired people along with the distance needs to travel while navigating through a supermarket. The algorithm can be used with any obstacle detection technique like ultrasonic sensors to reroute the user through the optimal path in case of obstacle. The performance of the algorithm is compared with that of Dijkstra’s shortest path algorithm and it is found that the proposed algorithm finds a better path in accordance with visually impaired people by reducing the number of turns in the paths and rerouting the user in case of obstacles.