**Dijkstra and A\* Algorithm**

Ahmed Saif K21-3232

Saleh Shamoon K21-3333

Huzaifa Quaid Johar K21-3272

**Problem:**

The path finding problem has been in our curriculum since the beginning of our studies and each new year we learn of a new method to overcome this problem. This time as we were tasked to create path finding algorithms with visualization, we will be doing so using the Dijkstra and A\* algorithms.

**Methodology:**

As mentioned above, we will be tackling this problem using Dijkstra and A\* algorithm. What these two specific algorithms are is defined below:

Dijkstra's algorithm, named after Edsger W. Dijkstra, is a pivotal method in computer science used to find the shortest path between nodes in a graph. It explores paths from a starting node, updating the shortest known distances to other nodes as it progresses. The algorithm selects the nearest unvisited node at each step, examining its neighbors to refine distance estimates. It continues until all reachable nodes are visited or reaches the destination, providing the shortest path from the starting node. Crucially, it assumes non-negative edge weights in the graph for accurate results.

**A\* algorithm** is a widely used heuristic search algorithm in computer science. It efficiently finds the shortest path from a starting node to a goal node in a graph. It employs a heuristic function to estimate the cost from the current node to the goal, combining this estimate with the actual cost to determine the most promising path. A\* explores paths by considering the total estimated cost, moving toward the goal node while continually updating and prioritizing the most efficient path.

**Language Used:**

We will be presenting our project using Python language alone, as upon our request you allowed us the privilege to do so.

**Expected Output:**

The expected output for both Dijkstra's and A\* algorithms is the shortest path from a starting node to a specific goal node (or all nodes in the case of Dijkstra's algorithm without a specific goal). Additionally, both algorithms provide the total cost or distance associated with the shortest path.

**Dijkstra's Algorithm Output:**

- Shortest distance from the starting node to all other nodes in the graph.

- For a specific goal node: the shortest path from the starting node to that goal node and the total distance/cost of that path.

**A\* Algorithm Output:**

- Shortest path from the starting node to a specific goal node.

- The total cost or distance of the found shortest path.

Both algorithms aim to provide optimal paths, with Dijkstra's algorithm exploring all reachable nodes to find the shortest paths from the starting node to every other node, while A\* focuses on finding the shortest path to a specific goal node by intelligently prioritizing based on a heuristic function.