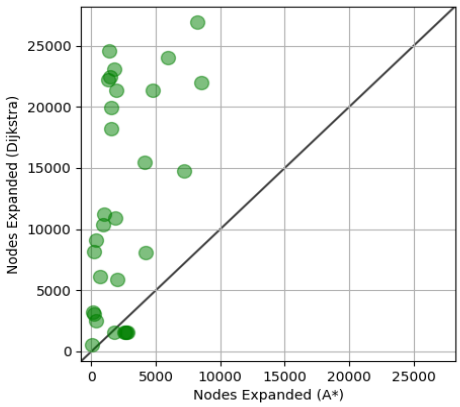
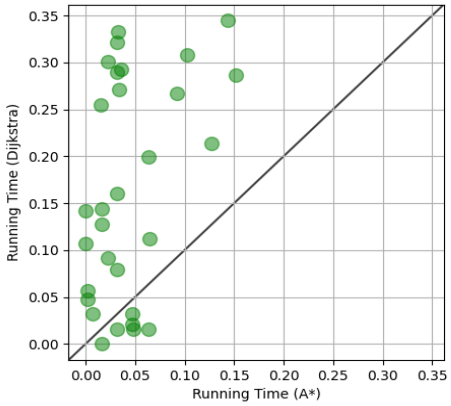
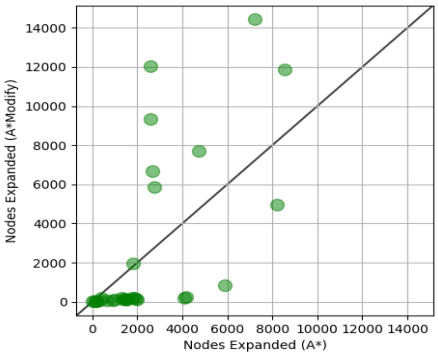
Answer 1) (a) In nodes expanded and in time taken majority of the points lie above the diagonal suggesting that A* expands fewer nodes and takes less time to reach the goal state. The reason for this is that A uses a heuristic to guide its search making to focus on the states that are near the goal meanwhile Dijkstra's algorithm explores all possible paths

(b) In the running time graph the points are more closer to the diagonal as compared to the explored nodes graph. This is because even though A* expands a fewer nodes, it runtime is more as compared to the number of nodes it expands (evident because in the time take graph, it is more towards the diagonal) .This is because the each expanded node involves additional work to compute and evaluate the heuristic. This added complexity can cause the running time of A* to be higher





Answer2) We observe that Modified A*(the one in which we multiply the heuristic with 2) expand more nodes than in normal A*. This is because the search becomes more greedy , it may follow a misleading path that appears promising due to overestimated heuristics. Which makes the algorithm to backtrack later on and explore additional nodes to find the right path and this increase the number of expansions in the node



Answer 3) Once a node is visited , its cost and parent remains the same. So if we discover the same node with a lower cost (a better path) then we wont be able to update the path. Hence, we will be stuck with a suboptimal solution or in some cases might not even find solution. As a result, the cost of the path returned will be more than the optimal path cost. In Astar this issue causes even more trouble because depending on the choice of heuristic it might lead to initially promising but eventually non-optimal sol , and if we are not able to update then we will be stuck with sub optimal solution.