1. If the Turing test is passed does this shows that computer exhibits intelligence? State your reasons.

In my opinion, if the Turing test is passed it doesn’t shows that computer exhibits intelligence. The computers now a days are able to have proper conversation with human that make us doubt “if that really is a computer “. But the computer replies based on data that is stored. The computer recognize and converts the speech of a person to test. Then it uses efficient search algorithm to find the reply of the input. And the reply that is found by computers is again translated to speech. Here we are using speech recognition, NLP, searching and text to speech only. If we ask question to the computer which is outside the domain then computer will fail to give correct answer. So if machine passes Turing test then it doesn’t mean they exhibits intelligence.

1. Given the following 8 puzzle. Define the problem as a search problem in terms of states. Operate a goal test and path cost.

|  |  |  |
| --- | --- | --- |
|  | 4 | 5 |
| 6 | 1 | 8 |
| 7 | 3 | 2 |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 8 |  | 4 |
| 7 | 6 | 5 |

The puzzle consists of 9 tiles each tiles is numbered from 1 to 8 and one is empty tile. Empty tile can be replaced by a numbered tile which is next, before, down or up to it. If the empty tile is at the centre then 4 moves are possible. If the tile lies at the corner then 2 moves are possible else 3 moves are possible.

We use A\* search algorithm to solve such kind of problem. We define a state of game to the board position. The number of moves made to reach the board position and previous state. First insert the initial state into the priority queue. Then delete from priority queue. The state with minimum priority and insert all neighboring state into priority queue. Repeat this procedure until we reach the goal state. The success of this approach depends upon the choice of priority function for the state

The following priority function are taken into consideration

1. Hamming Priority: This is sum of total no of blocks in wrong position . We assume a state with a small number of block position is close to the goal state, and we prefer a state that have been reached using small number of moves.
2. Manhaton Priority Function:- It is calculated by the sum of distance from block to their goal position plus the number of moves made so far to get to the state.

h2 = 2 +3 +3 +2 +2 +0 +2 = 16

The solution of this problem is determined by quality of a heuristic effective branching factor.

1. When will BFS outperform DFS? When will DFS outperform BFS? Compare.

BFS will outperform DFS on sparse graph with low branching factor where DFS can get stuck following long irrevlent chains.

DFS will outperform BFS on thickly connected graphs with high branching factor because BFS experiences an exponential blowup in nodes expanded as the branching factor increases.

BFS will use more memory depending on the branching factor. However, BFS is complete algorithm. The space and time complexity of BFS is O(bd+1)

DFS is going to use less memory because it only stores the path from root to leaf node beside the sibling of each node on the path and DFS removes a node from memory once all of it descendents have been expanded its time complexity is O(dd+1) and space complexity is O(bm)