**Assignment 1**

**Implemetation of A\* Algorithm**

**B.Tech CSE VII Semester, Winter 2018**

**Co-ordinator: U. A. Deshpande**

**Date: 7th August 2018**

**Hard Deadline: 16th August 2018 (R3, R4 - 1.30 pm and R1, R2 – 2.00 pm onwards)**

**Submit on email id:** [**uad.ai.assignment@gmail.com**](mailto:uad.ai.assignment@gmail.com) **with roll no, name and assignment number in subject line. (Viva in CSE 105 – contact Mansi A. Radke)**

Consider a path optimization problem with resources and locks. We are given a weighted undirected graph, and the goal is to travel as cheaply as possible from a given start vertex to a given goal vertex. Here the problem is that some of the vertices are blocked, and can only be unblocked if the agent is carrying an appropriate resource (key), which can be found in other vertices of the graph (which may also be blocked or not). A key is intended to act on a lock with the same ID. An agent at a vertex automatically picks up all the keys at that vertex just before starting the next move. Entering a locked vertex is only possible when the agent has the appropriate key, at which point the lock and key both disappear. An agent can do a traverse action from one node to another whose cost is equal to the edge weight connecting the two nodes. The action always succeeds if the target vertex is unblocked (or becomes unblocked due to having the correct key(s)), and fails otherwise. For a failed action, the agent stays at the same location but still pays the cost of traversal. Implement an agent using A\* search that picks the move with best immediate heuristic value. Heuristic function is of your choice.

Example graph could be as follows: S is source and T is goal node or target node. Nodes S, A, E, C, F, T are unblocked. Nodes B and D are blocked. E contains key for node B and D. To keep the problem simple, let us assume that a key for a node could be present with multiple nodes.

S A B C D E F T

S 9999 2 9999 9999 9999 3 9999 9999

A 2 9999 4 9999 5 7 9999 9999

B 9999 4 9999 25 2 9999 9999 5

C 9999 9999 25 9999 9999 9999 9999 9999

D 9999 5 2 9999 9999 5 4 8

E 3 7 9999 9999 5 9999 10 9999

F 9999 9999 9999 9999 4 10 9999 6

T 9999 9999 5 9999 8 9999 6 9999

Points to ponder: What should be the heuristic function? How to differentiate between a blocked and unblocked node? How to differentiate between a node with multiple keys and one with no keys or less keys? Is your heuristic function admissible?