10/11/2020

AI LAB TEST - 1

K. ARUN TEJA
1BM18CS041
5A

1.) 8 - puzzle (A* algorithm)

```
// heuristic function

def heuristic_f (self, start, goal):
        return self.heuristic ( start.data , goal ) + start.level

def heuristic (self, start, goal):
    temp = 0
    for i in range (0, self.n):
        for j in range ( 0, self.n):
            if ( start[i][j] != goal[i][j] and start[i][j] != '_'):
                temp += 1

    return temp

def process (self):
    print ("Enter start state : ")
    start = self.accept()
    print ("Enter goal state : ")
    goal = self.accept()

    start = Node (start ,0,0)
    start.fval = self.heuristic_f (start, goal)
    start.open.append(start)
    while True:
        cur = self.open[0]
        for i in cur.data :
            for j in i :
                print (j)
            print ('')
        if ( self.heuristic ( cur.data , goal ) == 0 ):
            break
```

// accept() is a function that accepts the matrix. (8-puzzle)

```
for i in cur.generate_child():
    i.fval = self.heuristic_f(i, goal)
    self.open.append(i)
self.closed.append(cur)
del self.open[0]

self.open.sort(key=lambda x: x.fval, reverse=False)
```

// generate_child() generates children (in 4 directions) from the given node

// Node function to generate each node in 8-puzzle

// main function.