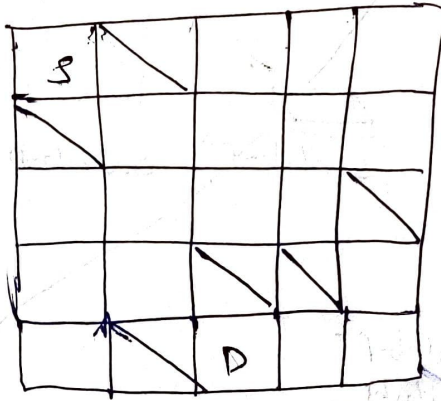


AI - Lab Test 1Program 4 :-A\* Search with euclidean distance :-Pseudocode :-

```
def f(curr-state, goal-state):
```

$$h = \sqrt{(\text{curr-state.x} - \text{goal-state.x})^2 + (\text{curr-state.y} - \text{goal-state.y})^2}$$

```
    return h
```

```
def a-star-search:
```

```
    def astarsearch(grid, src, dest):
```

```
        if isValid(src.first, src.second) == false:
```

```
            return
```

```
        if isValid(dest.first, dest.second) == false:
```

```
            return
```

```
        if isDestination(src.first, src.second, dest):
```

```
            return grid # Already at destination
```

```
        x = grid.x; y
```

```
        y = grid.y
```

```
        eclid = f(grid, dest)
```

if grid[x][y] == 2:

print "found at xd, yd", (x, y)

return True

elif grid[x][y] == 1:

print "wall at xd, yd", (x, y)

return false

elif grid[x][y] == 3:

print "visited xd, yd", (x, y)

return false.

#1 = wall #2 = available block 3 = visited block.

grid[x][y] = 3

if ((x < len(grid)-1) and asearch(x+1, y)) or  
(y > 0 and asearch(x, y-1)) or x > 0 and asearch(x-1, y) or

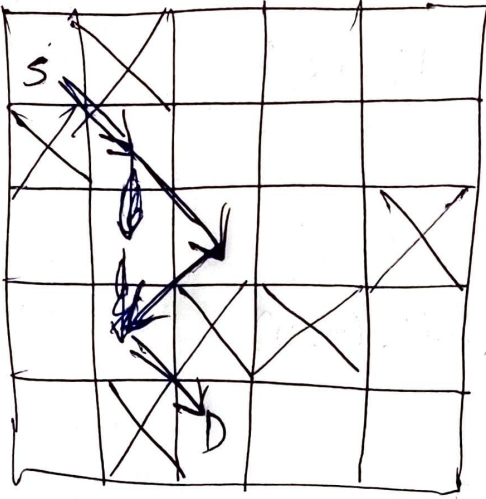
(y < len(grid)-1 and asearch(x, y+1)):

return True

return false.

asearch(0, 0).

Hence, both A\* search with euclidean function is included.



Traversal: -

$$(0,0) \rightarrow (1,1) \rightarrow (2,2) \rightarrow (3,1) \rightarrow (4,2)$$