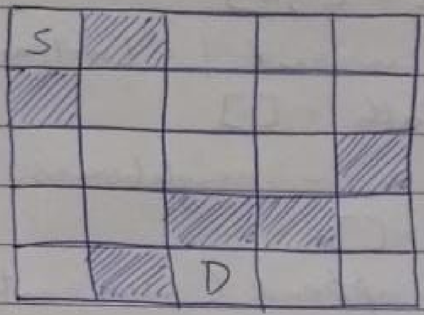
Anirudddh N. S
1BM18CS015
10/11/2020

AI Lab Test - 1

Q :- Implement A* algorithm using euclidean distance as heuristic, & solve the following maze structure for source & destination.



Solution :-

Using A* algorithm & euclidean distance with recursion.

~~def solve (self):~~

```
def euclidDistance (x, n, m):
    dist = math.sqrt ((n-1 - x[0])*2
                      + (m-1-x[1])*2)
    return dist

def findShortestPath (nextPath, n, m):
    minDist = 999
    next = []
    for x in nextPath:
        if (euclidDistance (x, n, m)
                < minDist):
            minDist = euclidDistance (x, n, m)
            next = n
        return next

def findPath(n, m):
    path.append([0,0])
    current = [0,0]
    while (current != [n-1, m-1]):
        nextPath = []
        for x in neighbours:
            a=[]
            a.append(current[0]+x[0])
            a.append(current[1]+x[1])
            if a[0]> -1 and a[0]<n
                    and a[1]> -1 and
                        a[1]<m:
                if (maze[a[0]][a[1]]):
                    if a not in path and a
                     not in closedPath:
                        nextPath.append(a);
        if (nextPath):
            current = findShortestPath(nextPath,
                                          n, m).
            path.append(current)
        else:
            if path:
                closedPath.append(current)
                path.pop()
              if path:
                  current = path[len(path)-1]
              else
                  print ("No path b/w them")
        else:
          print("No path is possible")
```