MAZE PROBLEM (USING A*)

```
import math
src=0,0
target=3,3
maze=[[0,1,0,0,1],[1,0,0,1,1],[1,0,1,0,1],[1,1,1,0,1]]

def h(pos1,pos2):
     x1,y1=pos1
     x2,y2=pos2
     return math.sqrt(pow((x1-x2),2)+pow((y1-y2),2))

def possible_moves(pos,visited):
    pos_moves=[]
    i,j=pos
    for l in [i-1,i,i+1]:
        for m in [j-1,j,j+1]:
            if  l>=0 and m>=0 and l<len(maze) and m<len(maze[0]) and
not((l,m)==(i,j)) and maze[l][m]!=1:
                 if (l,m) not in visited: pos_moves.append((l,m))
    return pos_moves

def search(src,target,visited,d):
    visited.append(src)
    if src==target: return visited
    pos_moves=possible_moves(src,visited)
    if(pos_moves==[]): return False
    scores=[h(x,target)+d for x in pos_moves]
    min_score=min(scores)
    selected_moves=[]
    for i in range(len(pos_moves)):
        if scores[i]==min_score: selected_moves.append(pos_moves[i])
    for move in selected_moves:
        if search(move,target,visited,d+1)!=False: return visited
    return False

def solve_maze(src,target):
    visited=[]
    res=search(src,target,visited,0)
    if not res: print('No path exists')
    else:
        print('Path :',res)
        display(res)


def display(moves):
    for i in range(len(maze)):
        for j in range(len(maze[0])):
            if (i,j) in moves: print('+',end=' ')
            else: print(maze[i][j],end=' ')
        print()
    print()

solve_maze(src,target)
```

OUTPUT

Test 1

```
maze=[[0,1,0,0,1],[1,0,0,1,1],[1,0,1,1,1],[1,1,1,0,1]]
src=0,0
target=3,3
solve_maze(src,target)

Path : [(0, 0), (1, 1), (1, 2), (2, 3), (3, 3)]
+ 1 0 0 1
1 + + 1 1
1 0 1 + 1
1 1 1 + 1
```

Test 2

```
src=0,1
target=2,3
maze=[[0,1,1,0,1],[1,0,0,1,1],[1,0,1,0,1],[1,1,1,0,1]]
solve_maze(src,target)

No path exists
```