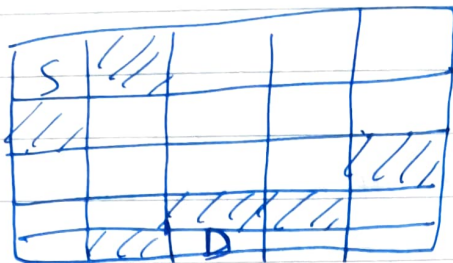


10/11/20

M...

AI LAB TEST 1

WRITE UP

Program / Pseudocode for the following Problem:

Euclidean distance:

```
def F(current_state, goal_state):
```

```
    h = sqrt((current_state.x - goal_state.x)
              **2 + (current_state.y - goal_state
                    .y)**2)
```

```
    return h
```

A star search:

```
def astarsearch(grid, srcsrc, destdest):
```

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

```
        # Invalid
```

```
        return
```

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

```
        # Invalid
```

```
        return
```

```
    if {isDestination(src.first, src.second), dest}
        == True:
```

```
        # Already at destination
```

```
        return
```

```
    h = x = src.x, y = grid.y
```

```
    euclid = F(grid, dest)
```

AI LAB TEST 1

WRITEUP

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

```
    print "found at %d,%d" % (x,y), return True
```

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

```
    print "Wall at %d,%d" % (x,y), return False
```

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

```
    print "Visited %d,%d" % (x,y), return False
```

Note: in our representation, 1=Wall, 2=destination, 3=visited cell

```
grid[1][1] = 3
```

```
if
```

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

```
    return True
```

```
    return False
```

```
astarsearch(0,0)
```

In the above writeup, Astar Search and Euclidean Functions are included.