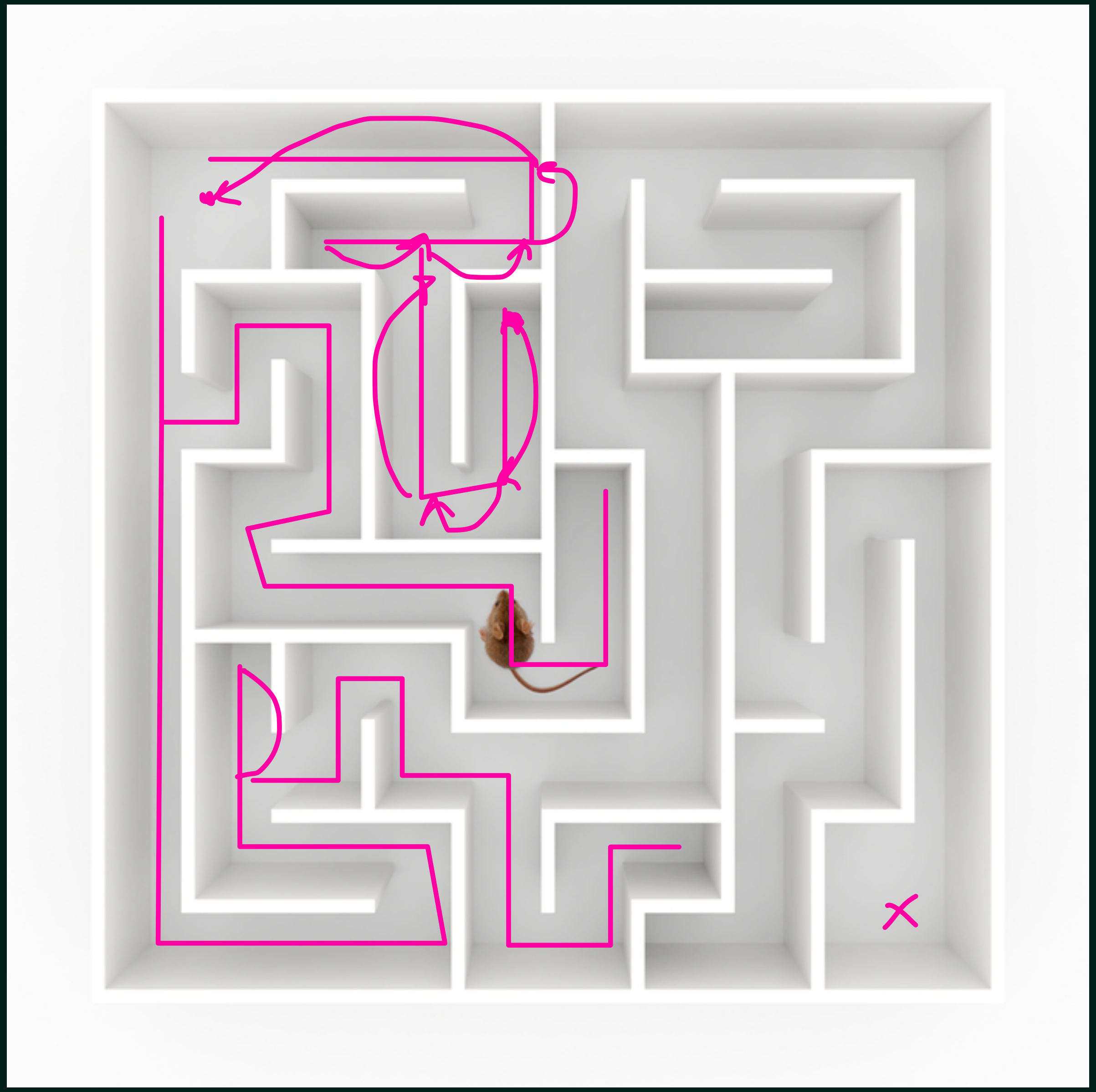
Backtracking - 1

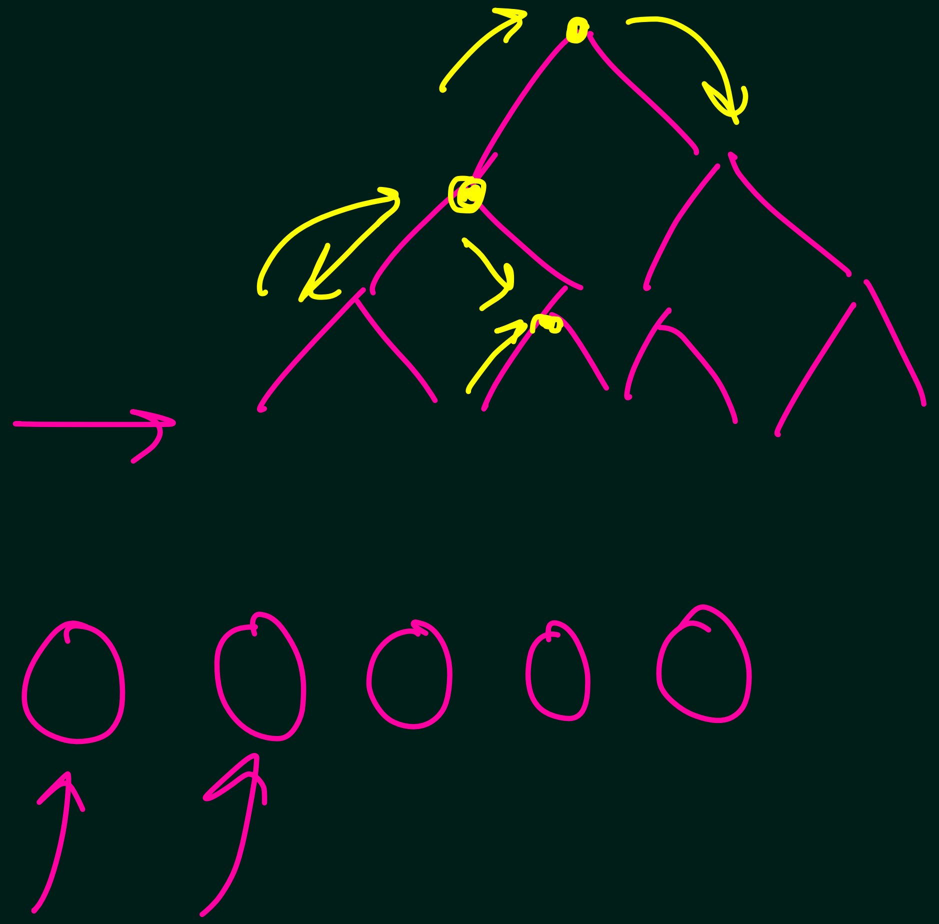
In This Lecture

1. What is Backtracking?
2. Rat in a Maze Problem ✓

What is Backtracking?

Backtracking is an algorithmic technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point in time.

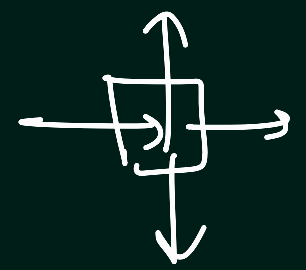


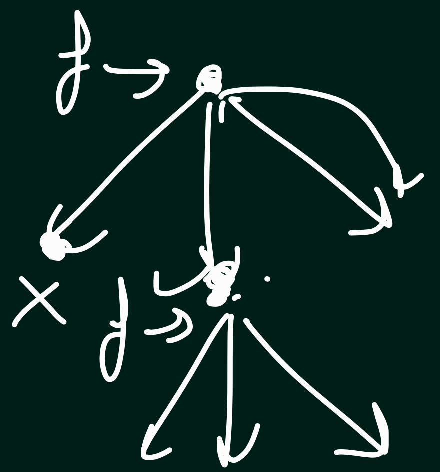


Common structure of Backtracking Solutions

```
boolean findSolutions(n, other params) :
    if (found a solution) :
        displaySolution();
        return true;

    for (val = first to last) :
        if (isValid(val, n)) :
            applyValue(val, n);
            if (findSolutions(n+1, other params))
                return true;
//          removeValue(val, n);
        return false;
```





```
f( ) {
    add()
    f( )
    remove() // backtracking.
}
```

Rat in a Maze Problem

Possible Directions: DLRU

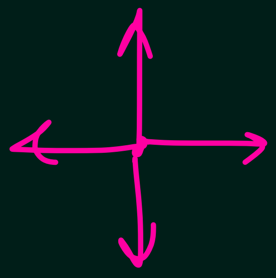
```
m[][] = {{1, 1, 1, 0},
         {1, 0, 0, 1},
         {1, 1, 0, 0},
         {1, 1, 1, 1}}
```

(i, j)

3 3

Output:

DDDRRR, DDRDRR



Recursion

① Subproblem

② substructure

① Base case → Destination

② D → L → R → U

③ Visited matrix

Rat in a Maze Problem

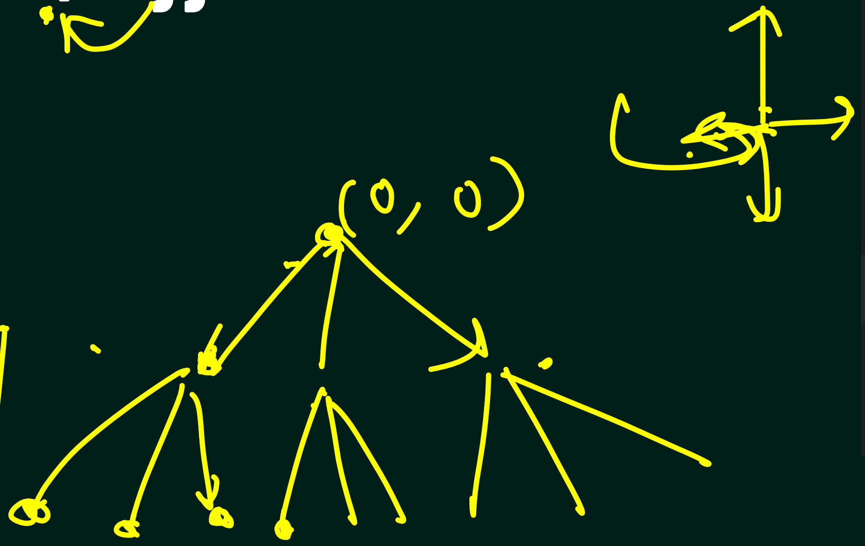
Possible Directions: DLRU

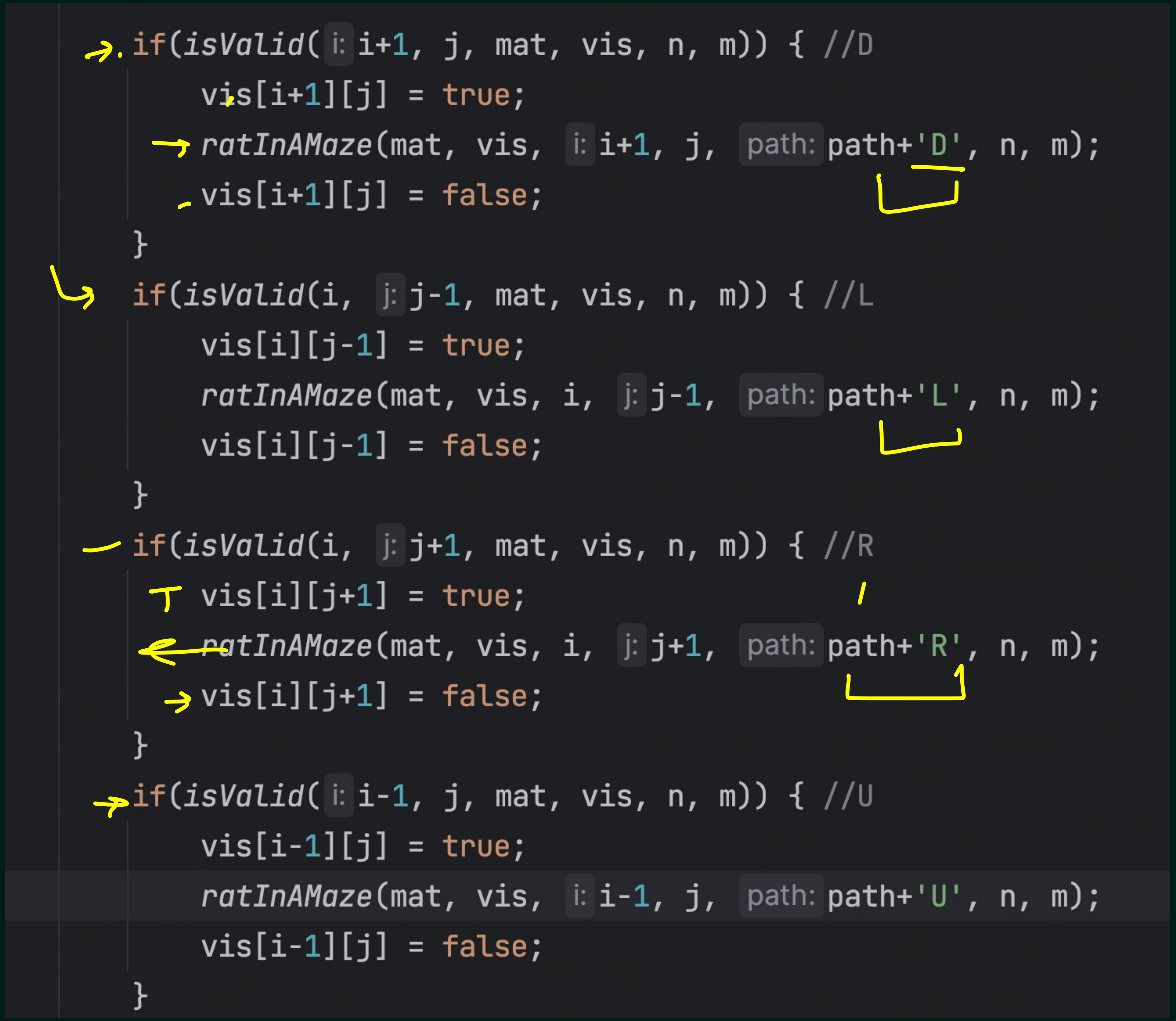
m[][] = {{1, 1, 1, 0},
{1, 0, 0, 1},
I D.D {1, 1, 0, 0},
DDD {1, 1, 1, 1}}

DDDRRR

T.C. $\boxed{O(3^{n*m})}$

S.C. $O(n^2)$





```
if(isValid( i: i+1, j, mat, vis, n, m)) { //D
    vis[i+1][j] = true;
    ratInAMaze(mat, vis,  i: i+1, j,  path: path+'D', n, m);
    vis[i+1][j] = false;
}
if(isValid(i,  j: j-1, mat, vis, n, m)) { //L
    vis[i][j-1] = true;
    ratInAMaze(mat, vis, i,  j: j-1,  path: path+'L', n, m);
    vis[i][j-1] = false;
}
if(isValid(i,  j: j+1, mat, vis, n, m)) { //R
    vis[i][j+1] = true;
    ratInAMaze(mat, vis, i,  j: j+1,  path: path+'R', n, m);
    vis[i][j+1] = false;
}
if(isValid( i: i-1, j, mat, vis, n, m)) { //U
    vis[i-1][j] = true;
    ratInAMaze(mat, vis,  i: i-1, j,  path: path+'U', n, m);
    vis[i-1][j] = false;
}
```