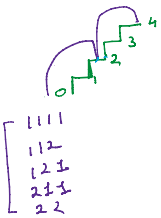
21st Feb

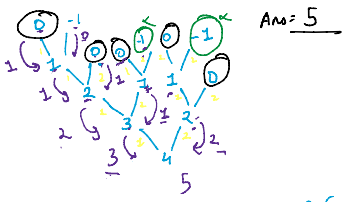
21 February 2022 18:56

Climbing Stairs :- https://leetcode.com/problems/climbing-stairs/

n = no. of stairs = 4

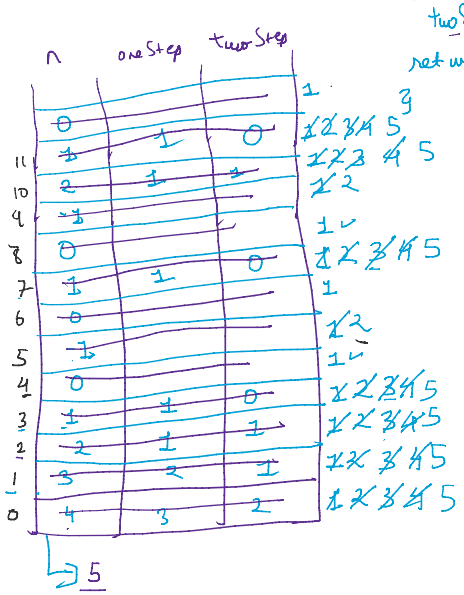
1 or 2 steps





Ans = 5

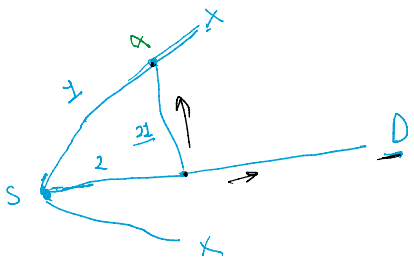
```
int CS(n) {
if(n<0) return 0;
if(n==0) return 1;
oneStep = CS(n-1)
twoSteps = CS(n-2)
return oneStep + twoSteps;
```



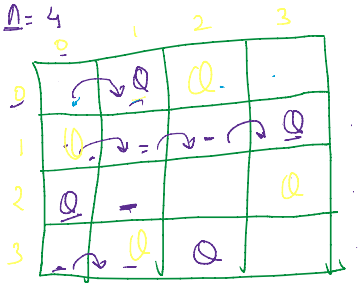
n = 4

```
public static int climbingStairs(int n) {
    if (n == 0) return 1;
    if (n < 0) return 0;
    int oneStep = climbingStairs( n: n - 1);
    int twoSteps = climbingStairs( n: n - 2);
    return oneStep + twoSteps;
}
```

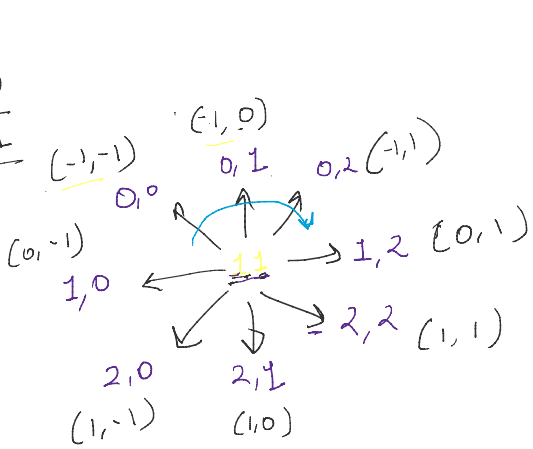
Backtracking



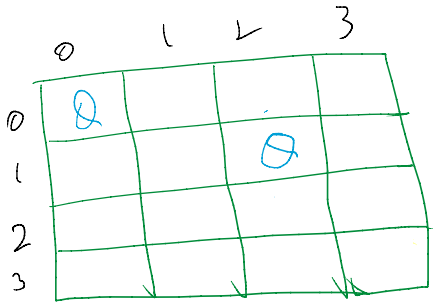
N-Queen



n x n



dir = [(-1, -1), (-1, 0), (-1, 1)]



nr = r + dirR
nc = c + dirC
dirR = -1
dirC = 0
i = 0 1

```java
public static boolean canBePlaced(char[][] ch
    int[][] dir = {{-1, -1}, {-1, 0}, {-1, 1}
    for (int i = 0; i < dir.length; i++) {
        int nr = r;
        int nc = c;
        int dirR = dir[i][0];
        int dirC = dir[i][1];
        while (nr >= 0 && nc >= 0 && nc < che
            if (chess[nr][nc] == 'Q') return
            nr += dirR;
            nc += dirC;
        }
    }
    return true;
}
```

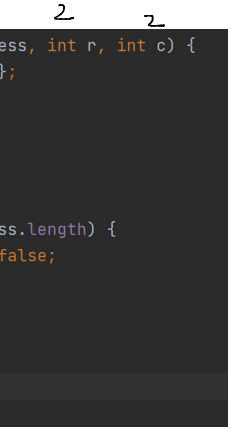
```
nQueen(chess, r, c,
if(r==queens) solFou
if(c==chess.length) re
if(isCanBePlaced(ches
    aess[r][c]='Q'
    nQueen(chess,
    chess[r][c]='.
}
nQueen(chess, r
}
```

Rat in A Maze :- https://practice.geeksforgeeks.org/problems/rat-in-a-maze-problem/1

```
    0  1  2  3
0 {{1, 0, 0, 0},
1  {1, 1, 0, 1},
2  {1, 1, 0, 0},
3  {0, 1, 1, 1}}
```

1 = traverse through it
0 = It is blocked

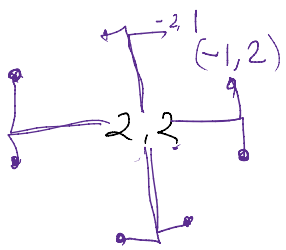
[DDRDRR
 DRDDRR



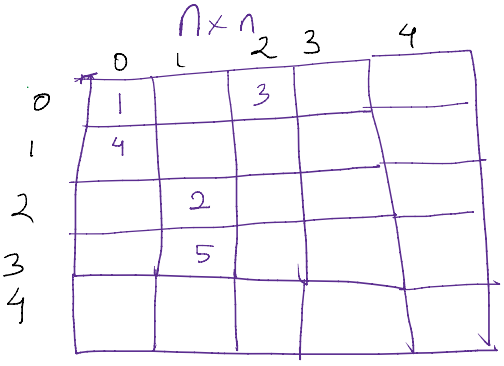
queens)
d ( ) :
turn :
, r, c)) {
:
r+1, 0, queens+1) ;
:
c+1, queens) ;

3 {0, 1, 1, 1}}

DR D D R R



Knights - Tour :-



=) Place the knight

=) Iterate over all directions
↳ if you can place the knight
(recursion

=) reset the knight