Recursion 4

Ques Find first index

0	1	2	3	4	5	6	7	8
4	1	6	2	6	2	4	1	7

⇒ 4 0

⇒ 1 1

⇒ 10 -1

Expectation

fi(arr, val, 0)

⇒ 0

Faith

fi(arr, val, 1)

⇒ 6

Combine

```
if (arr[idx] == val)
    return idx;
else return
    fi(arr, val, idx+1)
```

Ques fi(arr, val, arr.l - 1)

Last index

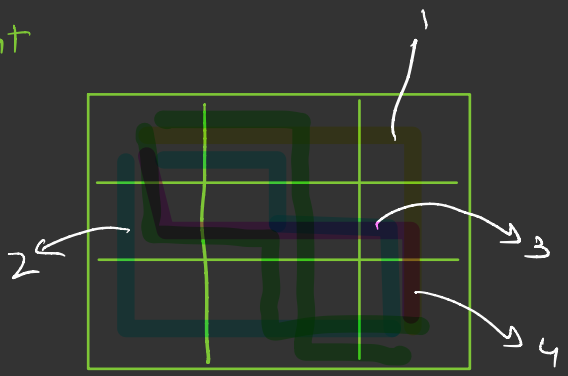
← Start in reverse

Ques. Maze Path Count

top Left (0,0) to bottom right

V (D)

H (R)



RRDD

DDRR

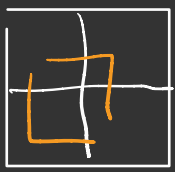
RDRD

DRRD

RDDR

DRDR

⇒ 6



⇒ 2

2 Directions ⇒ Right + Down

1 6	2 3	3 1
4 3	5 2	6 1
7 1	8 1	9 1

countMazePath(int arr[][], int r, int c)

Expectation

countMazePath(arr, 0, 0)

⇒ 6

Faith

countMP(arr, 0, 1)

↳ 3

countMP(arr 1, 0)

↳ 3

Combine

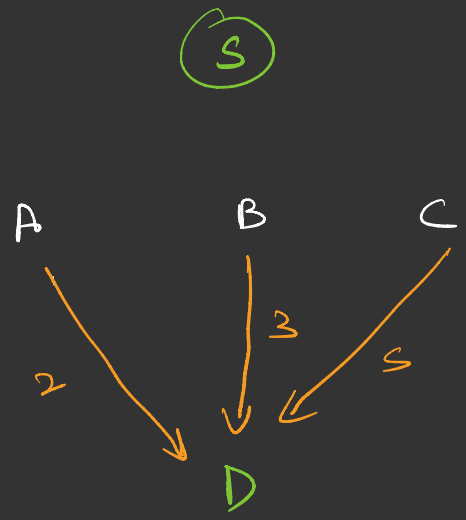
int qR = countMP(arr, 0, 1)

int qD = countMP(arr 1, 0)

return qR + qD;

countMP(arr, r, c+1)

countMP(arr, r+1, c)

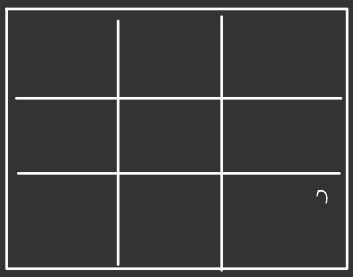


S - D

⇒ 2 + 3 + 5

⇒ 10

Ques Print Maze Paths



R R D D

R D R D

R D D R

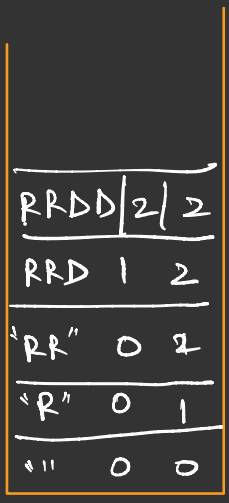
D R R D

D R D R

D D R R

① Pass psf in parameter

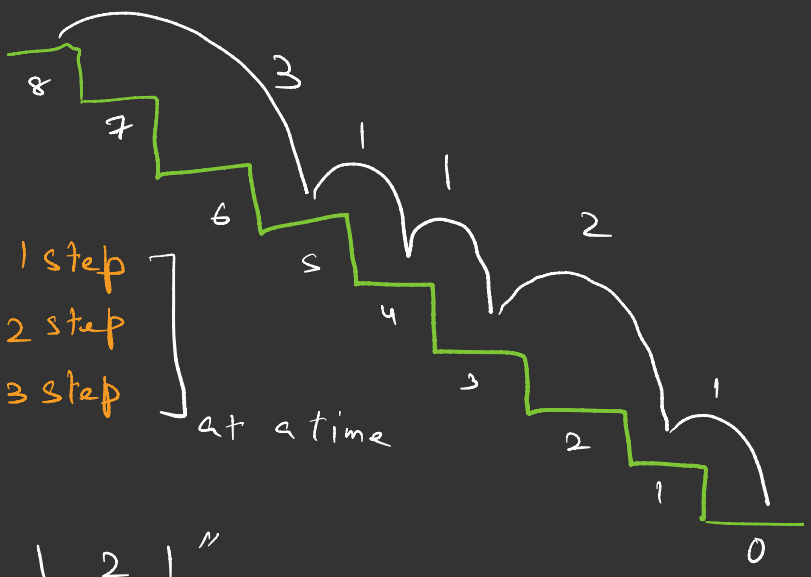
p R C



pMP (maze, psf+'R', r, c+1)

pMP (maze, psf+'D', r+1, c)

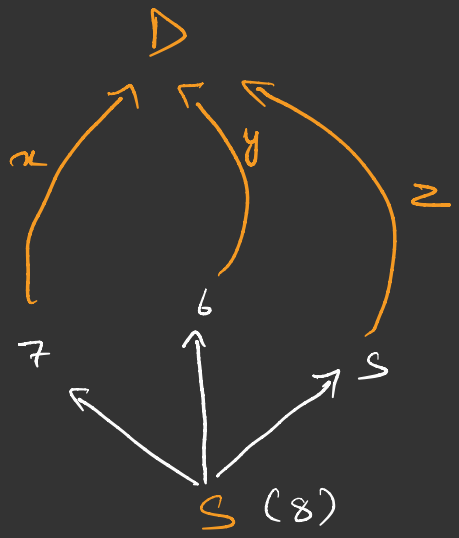
Ques Stair Paths



"3 1 1 2 1"

"2 2 2 2"

"2 3 1 2"



total ways from source to Destination

$x + y + z$