Assume : ① $N = 5$ $\quad 5 = 3\times 1 + 2\times 1$ or $2\times 1 + 3\times 1$ (can jump 2 or 3)

starting with 0th => $0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5$ one wa

② $N = 10$ $\quad 10 = 3\times 2 + 2\times 2$ or $2\times 2 + 3\times 2$

starting with 0th => $0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$

(There are 7 ways to do it)

Think, If we make an arry with the position zero to our target, all elements are zero

Key points : The frog can not land on 1 positions.
The frog can land on 2 position in one way
The frog can land on 3 position in one way
To Eeach step N, the frog could have come from step N-2 (by 2 step jump) or from N-3 (by 3 step jump)

because of that $N-2 \geq 0 \Rightarrow N \geq 2$ & $N-3 \geq 0$, $N \geq 3$

Use of a dyanamic array

Let, arr[0] → 1 *
arr[1] → 0 ← Never lands here
* { arr[2] → 1
arr[3] → 1 }

N=5

0	1	2	3	4	5
1	0	1	1	0	0

d = ← N+1 elements array

Frog starts Here

minimal landing points

The frog must reach the position d[5] using d[3] or d[2], so, $d[5] = d[3] + d[2] = 2$ ← two ways

1

For N^{th} position

$$d[N] = d[N-2] + d[N-3]$$ [Recursive funtions]

Algorithum

```
Function NO_of_ways(N):
    D = [0]* N
    D[0] = 1
    If N>=2:
        d[2] = 1
    if N >= 3:
        d[3] = 1
    For(int i=4; i<N+1; i++):
        d[i] = d[i-3] + d[i-2]

    return D[N]
```

have n+1 elements

Look at the code