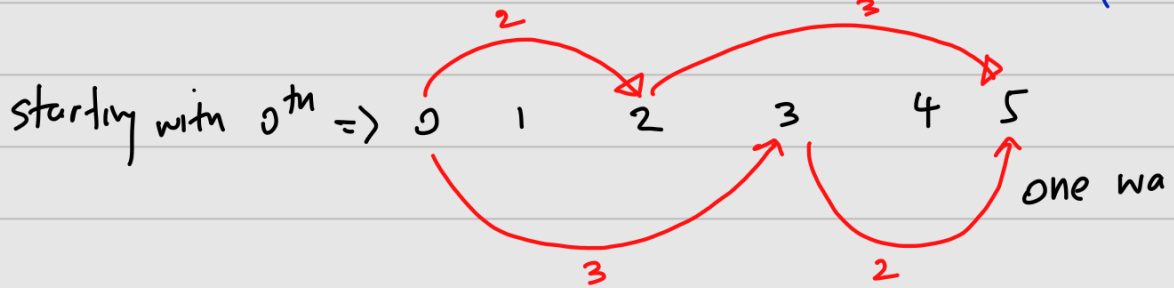
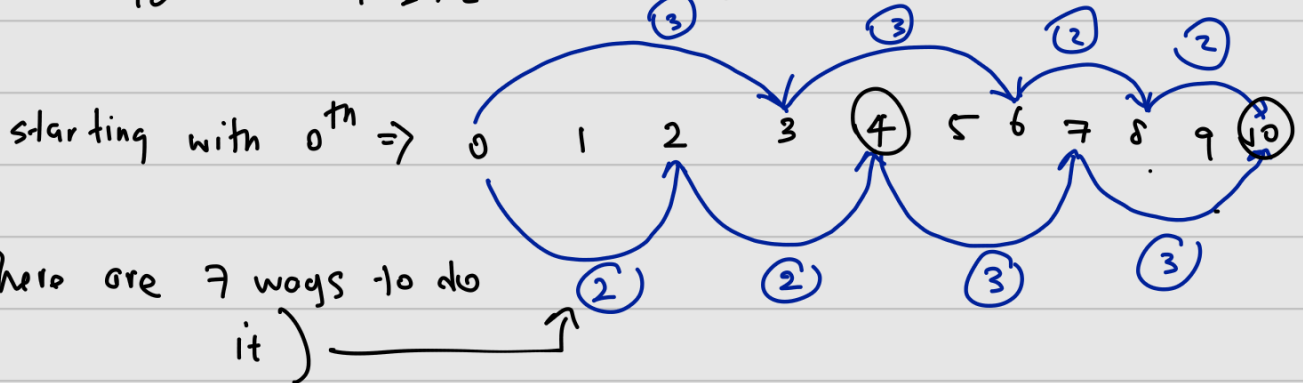


Assume: ① $N = 5$ $5 = 3 \times 1 + 2 \times 1$ or $2 \times 1 + 3 \times 1$ (can jump 2 or 3)



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



Think, If we make an array 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 Each 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 $N-2 \geq 0 \Rightarrow N \geq 2$ & $N-3 \geq 0 \Rightarrow N \geq 3$

Use of a dynamic array \rightarrow

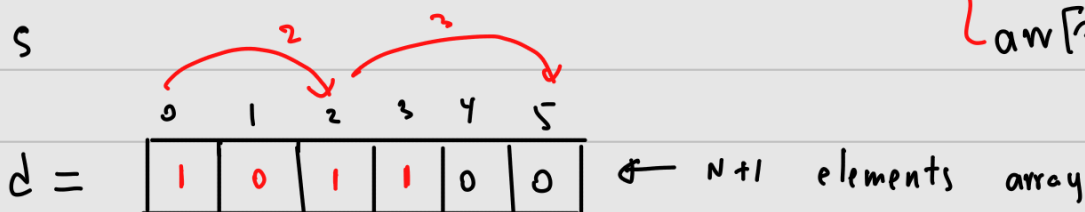
let, $arr[0] \rightarrow 1$

$arr[1] \rightarrow 0$ ← Never lands here

* $arr[2] \rightarrow 1$

$arr[3] \rightarrow 1$

$N = 5$



Frog starts here

minimal landing points

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

for N^{th} position

$$d[N] = d[N-2] + d[N-3] \quad [\text{Recursive Functions}]$$

Algorithm

Function $\text{NO_of_ways}(N)$:

$D = [0] * N$

$D[0] = 1$

If $N \geq 2$:

$d[2] = 1$

if $N \geq 3$:

$d[3] = 1$

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 