Runs Scored in every over

Runs	7	3	0	9	6	10	5
Over	1	2	3	4	5	6	7

25

$totalScore[i] = totalScore[i-1] + Runs[i];$

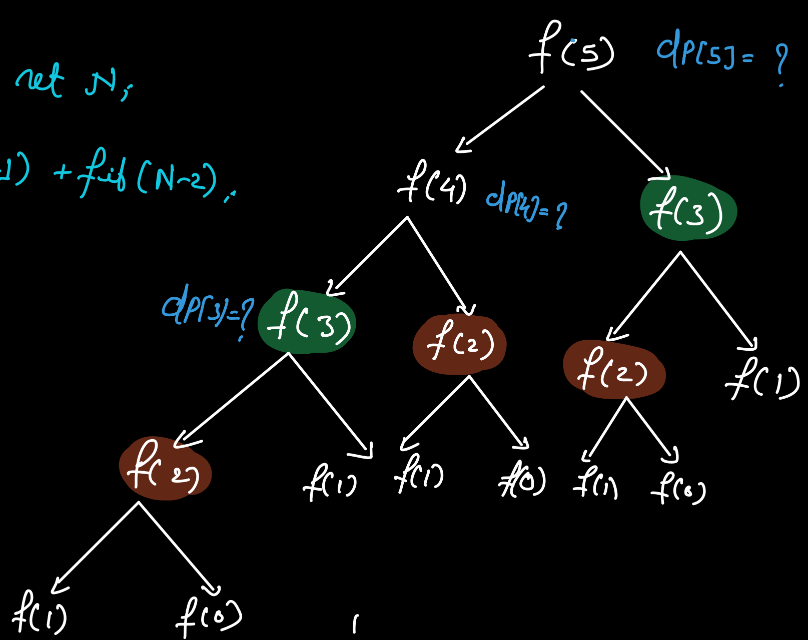
Prefix Sum

Those who do not remember the past are condemned to repeat it.

Dynamic Programming

	0	1	2	3	4	5	6	7	
fib :	0	1	1	2	3	5	8	13	. . .

```
int fib(N){
    if(N<=1) ret N;
    ret fib(N-1) + fib(N-2);
}
```

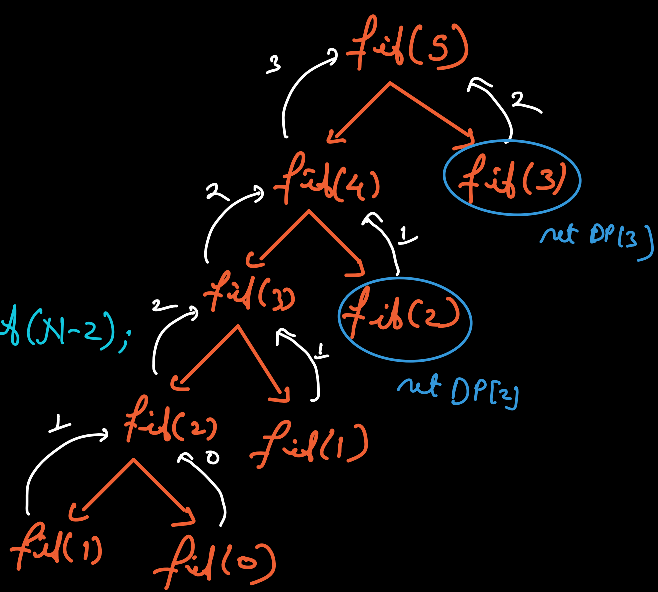


TC : $O(2^N)$
SC : $O(N)$

```
// dp[i] ----> ith fib no.
int dp[N+1] = {-1};

int fib( N) {
    if( N <= 1)
        ret N;

    if (dp[N] == -1){
        dp[N] = fib(N-1) + fib(N-2);
    }

    ret dp[N];
}
```

dp:

0	1	2	3	4	5
-1	-1	~~-1~~ 1	~~-1~~ 2	~~-1~~ 3	-1



TC : O(N)

SC : O(N) $(\underset{\text{dp}}{N} + \underset{\text{recursion stack}}{N})$

<u>DP with Memoization</u> (top - to - bottom)

Recursion ----> Optimal substructure

Duplicacy in recursion tree ----> Overlapping sub problems.

```
int fib(N){
    int dP[N+1];
    dP[0]=0;  dP[1]=1;
    for(i=2; i<=N; i++){
        dP[i] = dP[i-1]+dP[i-2];
    }
    ret dP[N];
}
```

	a	b	c			
	0	1	2	3	4	5
dP:	0	1	1	2	3	5

TC : O(N)

SC : O(N)

Bottom-up DP

Tabulation

```
int fib(N){
    a=0;  b=1;
    for(i=2; i<=N; i++){
        c = a+b;
        a = b;
        b = c;
    }
    ret c;
}
```

0	1	2
a	b	c
0	1	1
1	1	2
1	2	3
2	3	5

⋮

TC : O(N)

SC : O(1)

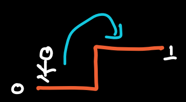
Amazon, MS, Adobe, Flipkart, Shopee, Zeta/Media.net, Myntra, Uber, Ola...

Q. Given N stairs. Count the no of ways of going from $0^{th} \longrightarrow N^{th}$ step.

Given that : from i^{th} step you can go to $(i+1)$ or $(i+2)^{th}$ step.

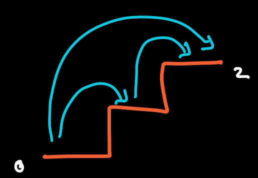
$N=0 \longrightarrow 1$

$N=1$



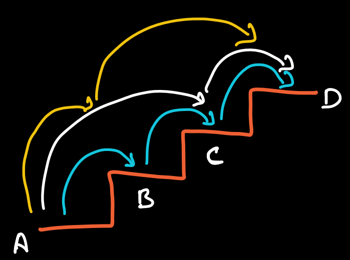
$\Rightarrow 1$

$N=2$



$\Rightarrow 2$

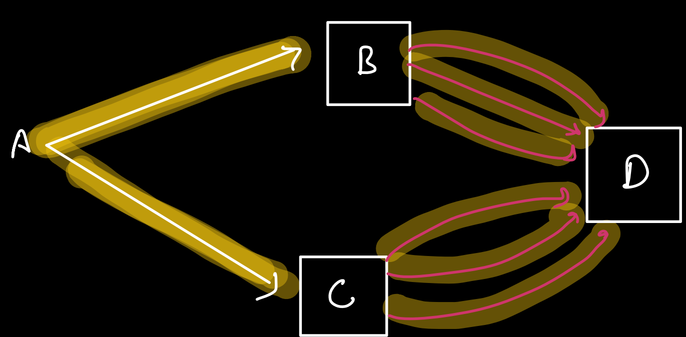
$N=3$



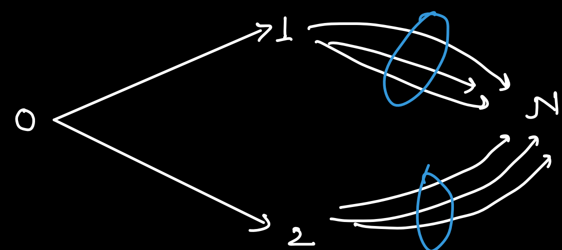
$\Rightarrow 3$

$(A\to D) = (A\to B) + (A\to C)$

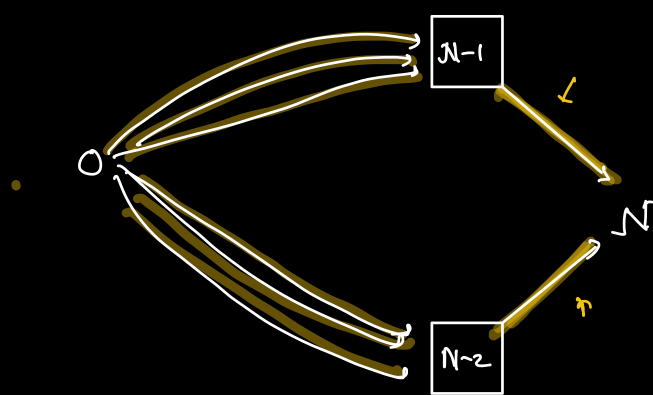
$1 + 2$



$\text{Paths}(A\to D) = \text{Paths}(B\to D) + \text{Paths}(C\to D)$



Paths[0→N] = Paths[1→N] + Paths[2→N]



Paths [0 to N] = Paths [0 to N-1] + Path [0 to N-2]

paths (N) ⟶ No of ways to reach N from 0

$$\boxed{\text{Path}(N) = \text{Path}(N-1) + \text{Path}(N-2)}$$

Q Given a one dice (6-faced) & a no N.
Count the no of ways to get a sum N if you can roll the dice as many times as req.

$N = 4 \longrightarrow 8$

1 (1, 1, 1
1, 2
2, 1
3)

2 (1, 1
2)

3 (1)

4 ( )

1 1 1 1
1 1 2
1 2 1
1 3

2 1 1
2 2

3 1

4

$N = 0 \longrightarrow 1$

$N = 1 \longrightarrow 1$

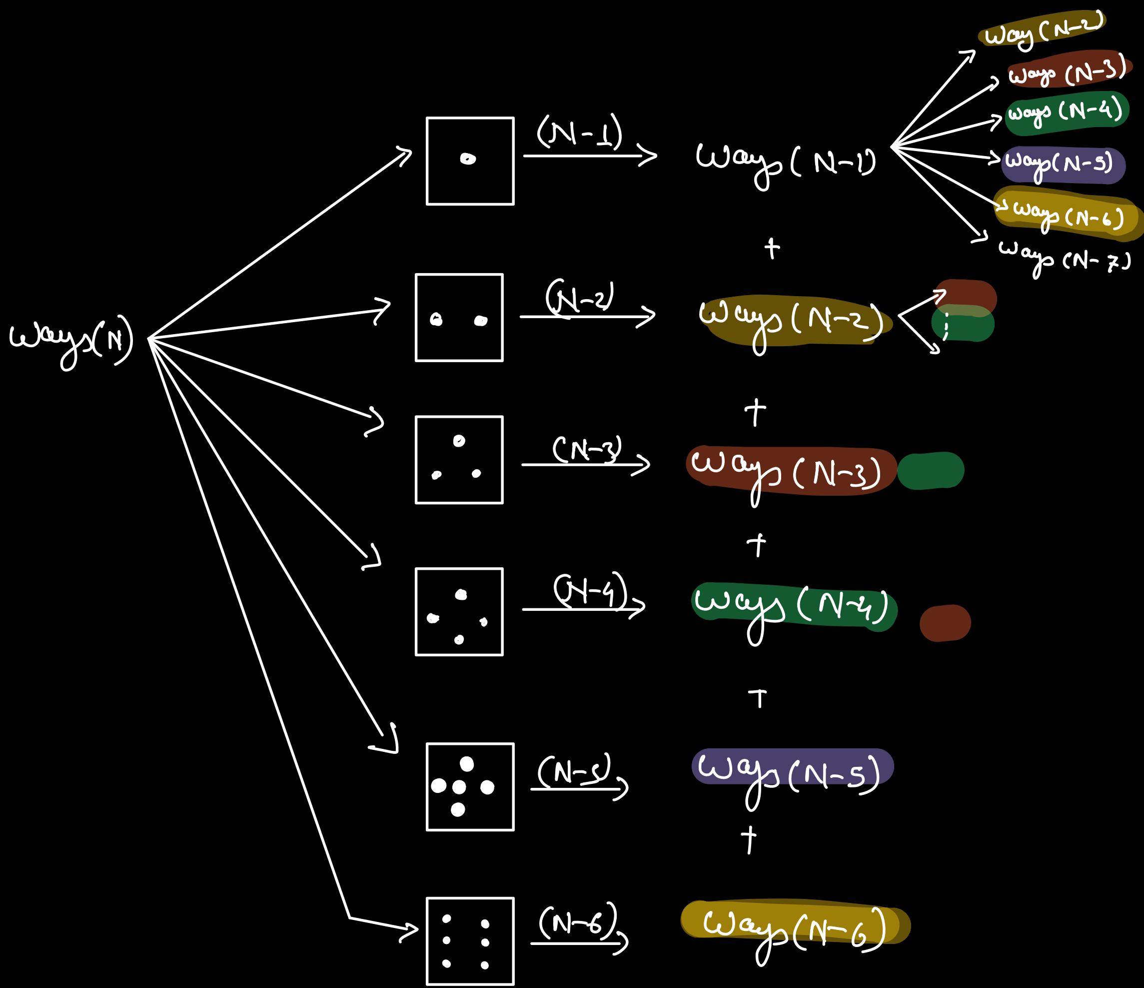
1

$N = 2 \longrightarrow 2$

1, 1
2

$N = 3 \longrightarrow 4$

1, 1, 1
1, 2
2, 1
3



$$dp(i) = dp[i-1] + dp[i-2] + dp[i-3] + dp[i-4] + dp[i-5] + dp[i-6]$$

$\boxed{\text{TC} = \text{No of function calls in the tree} \times \text{TC of one fn call (excluding the rec)}}$

TC = O(N) (N × O(1))

SC = O(N)

Gitesh

$dp[7] = \{0\}$

$dp[0] = 1$

$\vdots$

$\vdots$

→ Memoization

→ Bottom-up with O(N) Space

→ Optimize the space

Q **Coin Change I**

Unbnd

Given some coins with their denomination & a val K. Count the no of ways of getting the sum K by using coins. (One coin can be used multiple times).

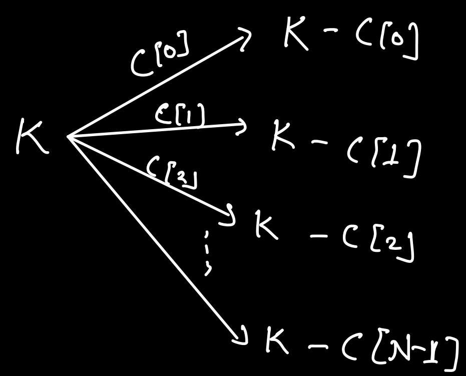
C : [1, 2, 5, 10]   K = 7

5, 2

1, 1, 1, 1, 1, 2

1, 1, 5

⋮



TC: O(NK)

Q Given a no K. Find the min no of perfect sq needed to get the sum K.

$N = 6$

$1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 \Rightarrow 6$

$\underline{1^2} + \underline{1^2} + \underline{2^2} \Rightarrow 6 \Rightarrow 3$

$1^2 + 2^2 + 1^2 \Rightarrow 6$

$2^2 + 1^2 + 1^2 \Rightarrow 6$

$N = 10$

$3^2 + 1^2 \Rightarrow 2$

$N = 9$

$3^2 \Rightarrow 1$

$dp(N) \xleftarrow{\min}$

$\xrightarrow{1^2} dp(N - 1^2)$

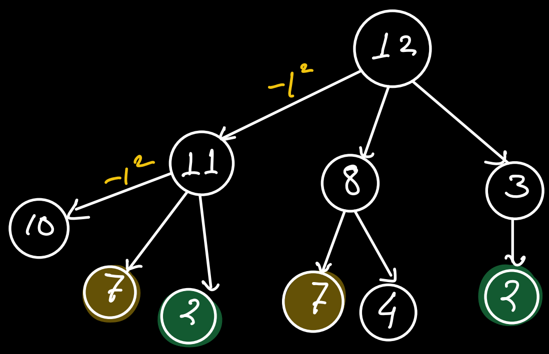
$\xrightarrow{2^2} dp(N - 2^2)$

$\xrightarrow{3^2} dp(N - 3^2)$

$\xrightarrow{4^2} dp(N - 4^2)$

$\vdots$

$\xrightarrow{(\sqrt{N})^2} dp(N - (\sqrt{N})^2)$



TC: $O(N\sqrt{N})$