Runs scored in every over

Runs	7	3	0	9	6	10	5
Over	1	2	3	4	5	6	7
					25	35	40

totalScore[i] = totalScore[i-1] + Run[i]

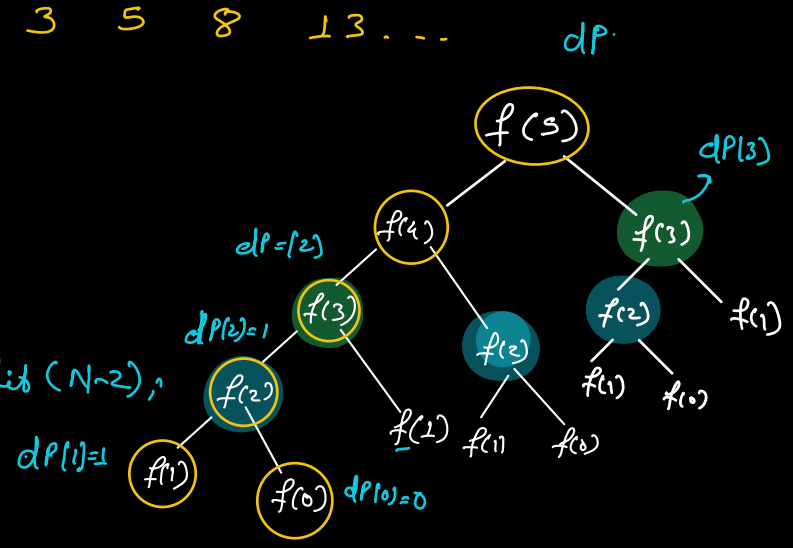
Those who do not remember the past are condemned to repeat it.

Dynamic Programming

fib : 0 1 1 2 3 5 8 13 . . .

(indexes: 0 1 2 3 4 5 6 7)

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



TC : $O(2^N)$

nodes in the tree (fn calls being made) : $O(2^N)$

unique fn calls : $O(N)$

Optimal sub-structure
→ Recursive eq

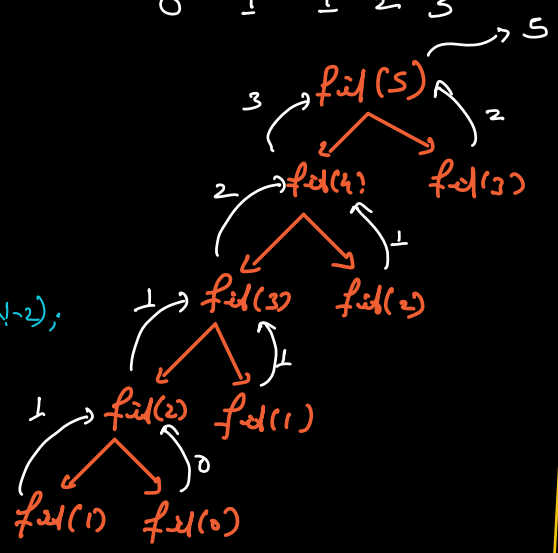
Overlapping sub-problems
→ Repetition of fn calls

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

TC: O(N)
SC: O(N)

dP:

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



Top-down DP

DP with memoization

Bottom-up / Iterative solution

(Tabulation)

0	1	2	3	4	5
0	1	1	2		

```
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];
}
```

TC: O(N)
SC: O(N)

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

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

TC : O(N)
SC : O(1)

Amazon, MS, Flipkart, Adobe, Shopee, Zeta/Media.net, clevertap, meta, Google
Uber, Ola ---

Q Given N stairs. Count the no of ways of going from

$0^{th} \longrightarrow N^{th}$ step

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

$N = 0$

$\longrightarrow 1$

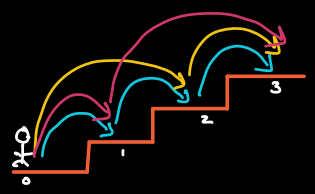
$N = 1$

$\longrightarrow 1 \quad \{0 \to 1\}$

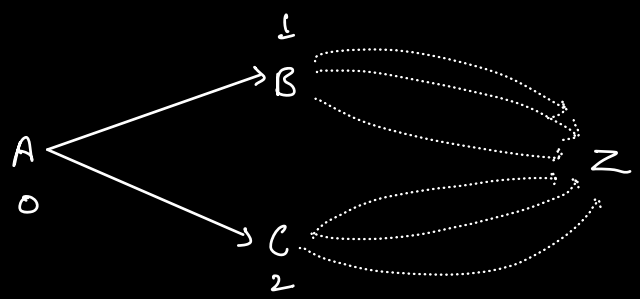
$N = 2$

$\{0 \to 1, \ 1 \to 2\}$
$\{0 \to 2\}$
$\longrightarrow 2$

$N = 3$



$\{0 \to 1, \ 1 \to 2, \ 2 \to 3\}$
$\{0 \to 2, \ 2 \to 3\}$
$\{0 \to 1, \ 1 \to 3\}$
$\longrightarrow 3$

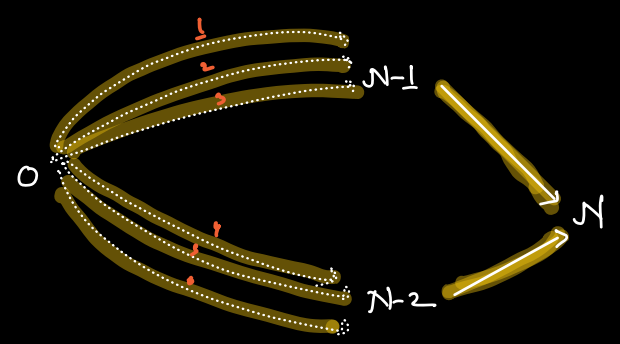


$\text{Paths}[A \to Z] = \text{Paths}[B - Z] + \text{Paths}[C - Z]$

$\text{Path}(i) \longrightarrow$ No of ways to reach N^{th} step from i^{th} step.

$$Path(0) = Path(1) + Path(2)$$

$$Path(i) = Path(i+1) + Path(i+2)$$



$$Path[0 \to N] = Path[0-(N-1)] + Path[0-(N-2)]$$

$Path(i) \longrightarrow$ no. of ways of reaching i from 0

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

Q House Robber

Amazon
Tokopedia
Paypal
Goldman Sachs

$\Rightarrow 12$

Given an array (+ve). Return the max sum without any adjacent element.

Approach 1

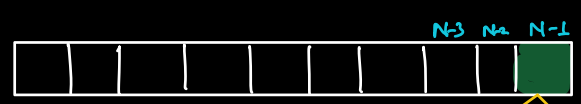
$$\max\begin{pmatrix}\text{Sum of even indexed elements}\\ \text{Sum of odd indexed elements}\end{pmatrix}$$

0	1	2	3	4	5
1	1	100	2	1	100

Σ even $= 102$

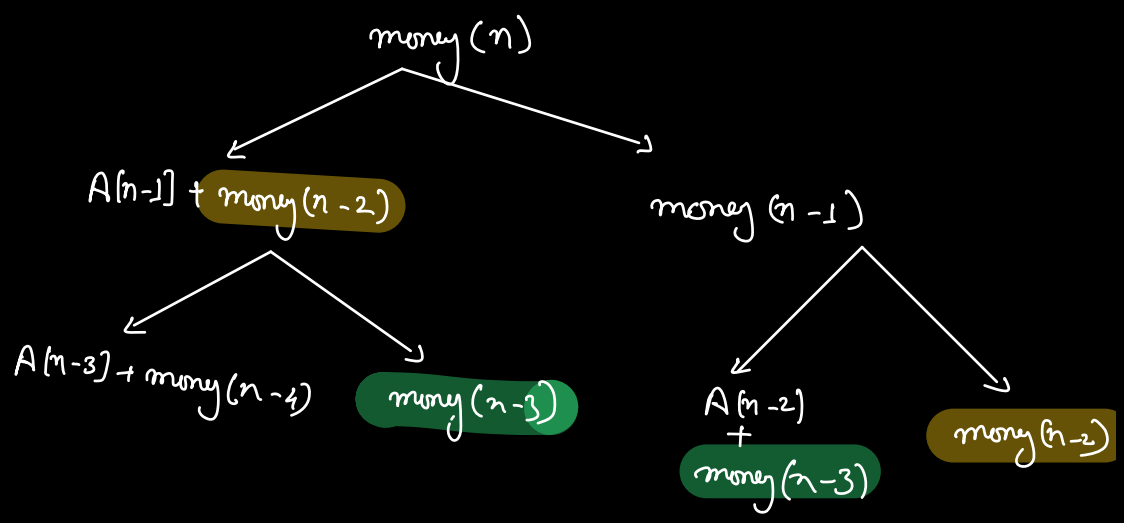
Σ odd $= 103$ Wrong!

ans $= 201$



$$\text{money}(N) = \max\begin{pmatrix}\text{rob}(N-1) & \text{skip}(N-1)\\ A[N-1] + \text{money}(N-3) & \text{money}(N-1)\end{pmatrix}$$

The max amount of money that can be robbed from the first N houses
index: $[0, N-1]$



Ex

0	1	2	3	4
2	7	9	3	1

0	1	2	3	4	5
0	2	7	11	11	12

dP[i]

no of houses that we have

$$dP[i] = \max\begin{cases} A[i-1] + dP[i-2] \\ dP(i-1) \end{cases}$$

$dP[2] = \max(7+0, 2) \longrightarrow 7$

$dP[3] = \max(9+2, 7) \longrightarrow 11$

$dP[4] = \max(3+7, 11) \longrightarrow 11$

$dP[5] = \max(1+11, 11) \longrightarrow 12$

- Do not miss the class
- Re-do the class stuff on your own (intuition)
- Solve assignments (implementation)
- Attempt the HW