DP 1

Good
Evening

Today's content

01. Introduction
02. Fibonacii
03. Stair ways (Amazon, Google)
04. Minimum squares to reach N (Walmart, Amazon)

{ 2 4 1 3 4 } $\Rightarrow 14 + 5 = \underline{19}$ 5

$$\boxed{pf[i] = pf[i-1] + ar[i]}$$

Fibonacii series

0 1 1 2 3 5 8 13 21 . ...

$$fib(n) = fib(n-1) + fib(n-2)$$

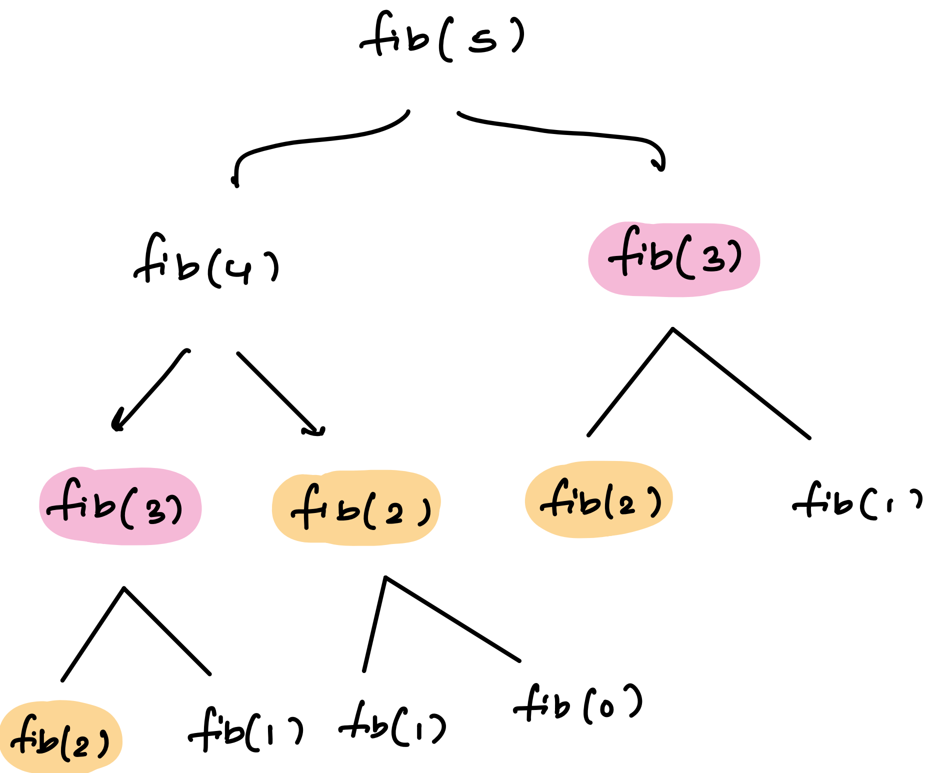
$fib(0) = 0 \qquad fib(1) = 1$

```
int fib(n)
{
    if (n ≤ 1) return n;

    return fib(n-1) + fib(n-2)
}
```

TC : $O(2^n)$

SC : $O(n)$

optimal substructure → solving problems by solving smaller subproblems



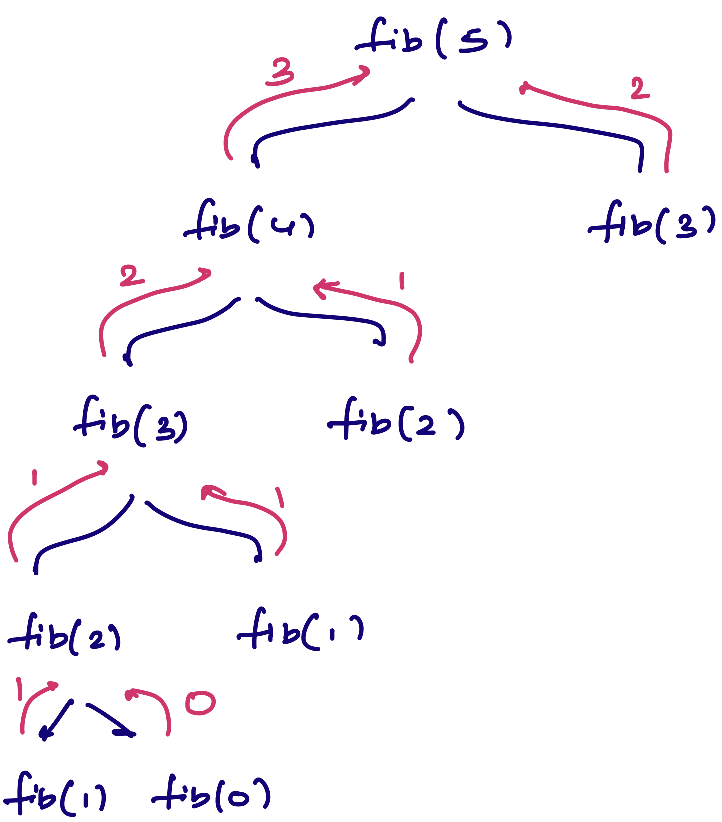
Overlapping subproblems → solving some problems multiple times

Solution → Storing the answer for already solved problem & reusing it again

n^{th} fibonaccii $\underline{0 \rightarrow N}$

int [] dp = new int [N+1] → initialise it with -1

```
int fib(n)
    if (n ≤ 1) return n;
    if (dp[N] != -1) return dp[N]
    dp[N] = fib(N-1) + fib(N-2)
    return dp[N];
}
```



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

TC : O(N)

SC : O(N)

N recursive calls

N dp array

Top-down Approach → start from the biggest problem

(recursion)

memoization

Bottom up Approach → start from the smallest problem

(Iterative)

Tabulation

fib(0) = 0

fib(1) = 1

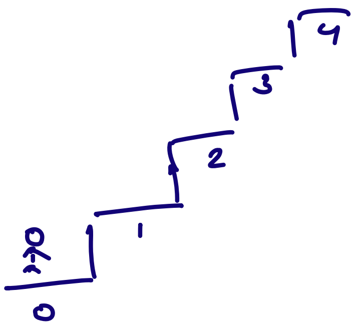
```
int dp[N+1]
  dp[0] = 0;
  dp[1] = 1;

for (i=2; i<=n; i++){
    dp[i] = dp[i-1] + dp[i-2];
}

return dp[N];
```

TC : O(N)

SC : O(N)

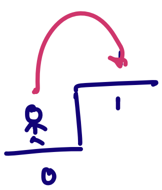
No recursive space

Stairs → No. of ways to reach N^{th} stair

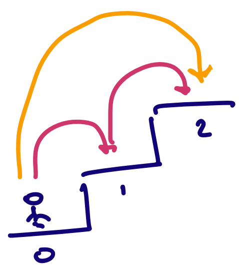


N stairs

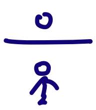
1 step   2 step



N = 1
ans = 1

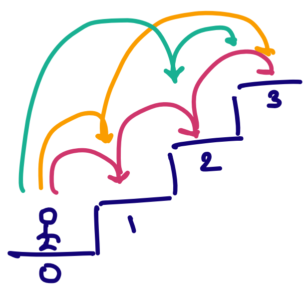


N = 2
ans = 2



N = 0
ans = 1

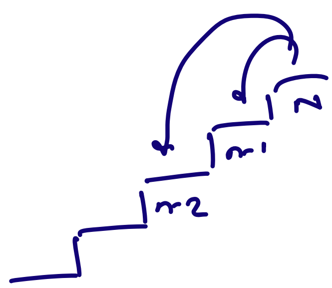
N = 3
ans = 3

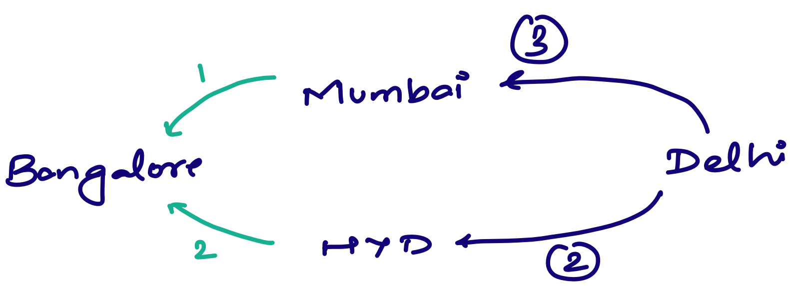


1 1 1
1 2
2 1

$n \xrightarrow{\text{1 step}} n-1$

$n \xrightarrow{\text{2 steps}} n-2$





⇒ 3*1 + 2*1

⇒ 5

5 step

Jump of 1 — Jump of 2

4 step — 3 steps

1 1 1 1
1 1 2
1 2 1
2 1 1
2 2

1

1 1 1
1 2
2 1

2

From 0 to 4 there are 5 ways, then from 0 to 5 as well there are 5 ways

From 0 to 3, there are 3 ways. then from 0 to 5, there are 3 ways

$$ways(n) = ways(n-1) * 1 + ways(n-2) * 1$$

$$ways(1) = 1 \qquad ways(2) = 2$$

DP[i] = No. of ways to reach i^{th} idx from 0^{th} idx

0	1	2	3	4	5
1	1	2	3	5	8
•	1	11	111	1111	
		2	21	211	
			12	121	
				112	
				22	

$$Dp[i] = Dp[i-1] + Dp[i-2]$$

$$Dp[0] = 1$$

$$Dp[1] = 1$$

10:07 pm → 10:17 pm

* Min no. of perfect sq

Find minimum no. of perfect squares required to get sum = N (duplicate squares are allowed)

$N = 6 \quad = 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2$

$= 2^2 + 1^2 + 1^2$, ans=3

$N = 10 \quad = 1^2 + 1^2 + \dots \quad = 10 \text{ times}$

$= 3^2 + 1^2 \quad = 2 \text{ no. req}$

ans = 2

$= 2^2 + 2^2 + 1^2 + 1^2 = 4 \text{ no. req}$

$= 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 = 7 \text{ no req}$

$\underline{\underline{N = 9}} \quad \underline{\underline{3^2}}$

ans = 1

$\boxed{N - \text{nearest perfect square}}$

~~greedy approach~~

$2^2 + 2^2 + 2^2 \rightarrow$ Ans = 3 ✓

$N = 12 \quad = 12 - 3^2$

$= 3 - 1^2$

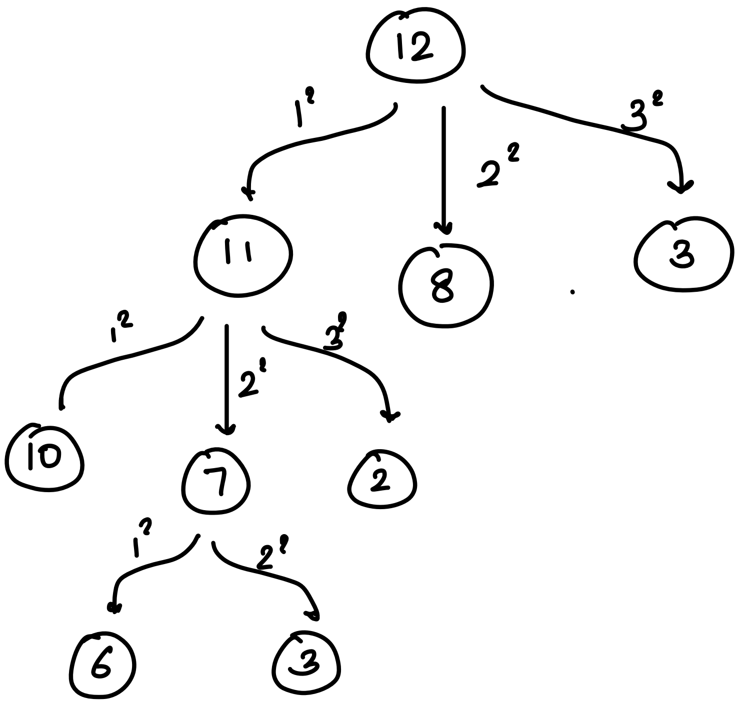
$= 2 - 1^2$

$= 1 - 1^2$

$= 0$

$\underline{\underline{\text{Ans} = 4}}$ ✗

Brute force → Try every possible way to form this sum.



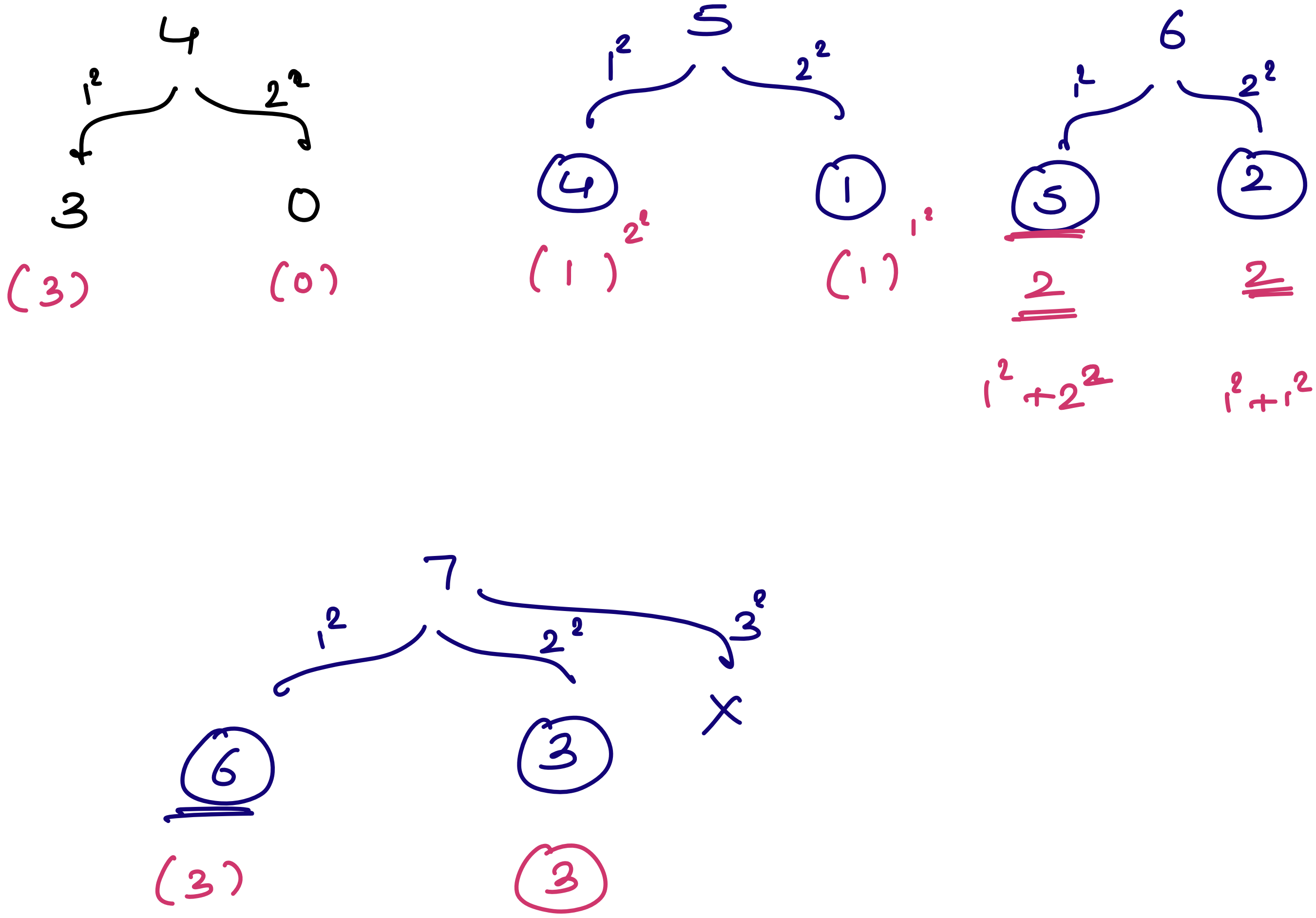
$$minsq(12) = 1 + min(minsq(11), minsq(8), minsq(3))$$

$$minsq(12) = 1 + min \begin{cases} minsq(12 - 1^2), \\ minsq(12 - 2^2), \\ minsq(12 - 3^2). \end{cases}$$

$$\begin{cases} minsq(i) = 1 + min\left(minsq(i - x^2) \quad \forall\, x^2 \leq i \right\} \end{cases}$$

```
int dp[N+1];

int psquare ( int N, int [] dp)                    ans=∞
    if (n==0) return 0;
    if (dp[N] != -1) return dp[N];
        ans= ∞ ;
    for (x=1 ; x*x ≤ n ; x++) {
        ans= min ( ans, psquare ( N-x^2 , dp)
    }
    dp[N] = ans +1 ;
    return dp[N];
}
```

0	1	2	3	4	5	6	7
0	1	2	3	1	2	3	4
	1^2	$1^2 + 1^2$	$1^2 + 1^2 + 1^2$	2^2	$1^2 + 2^2$	$1^2 + 2^2 + 1^2$	$1^2 + 2^2 + 1^2 + 1^2$




```
dp[0] = 0;

for(i=1 ; i<=N ; i++){
        ans = ∞
    for(x=1; x*x <= i ; x++){
        ans = min(ans, dp[i-x²]);
    }
    dp[i] = ans + 1;
}

return dp[N]
```

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

SC: $O(N)$