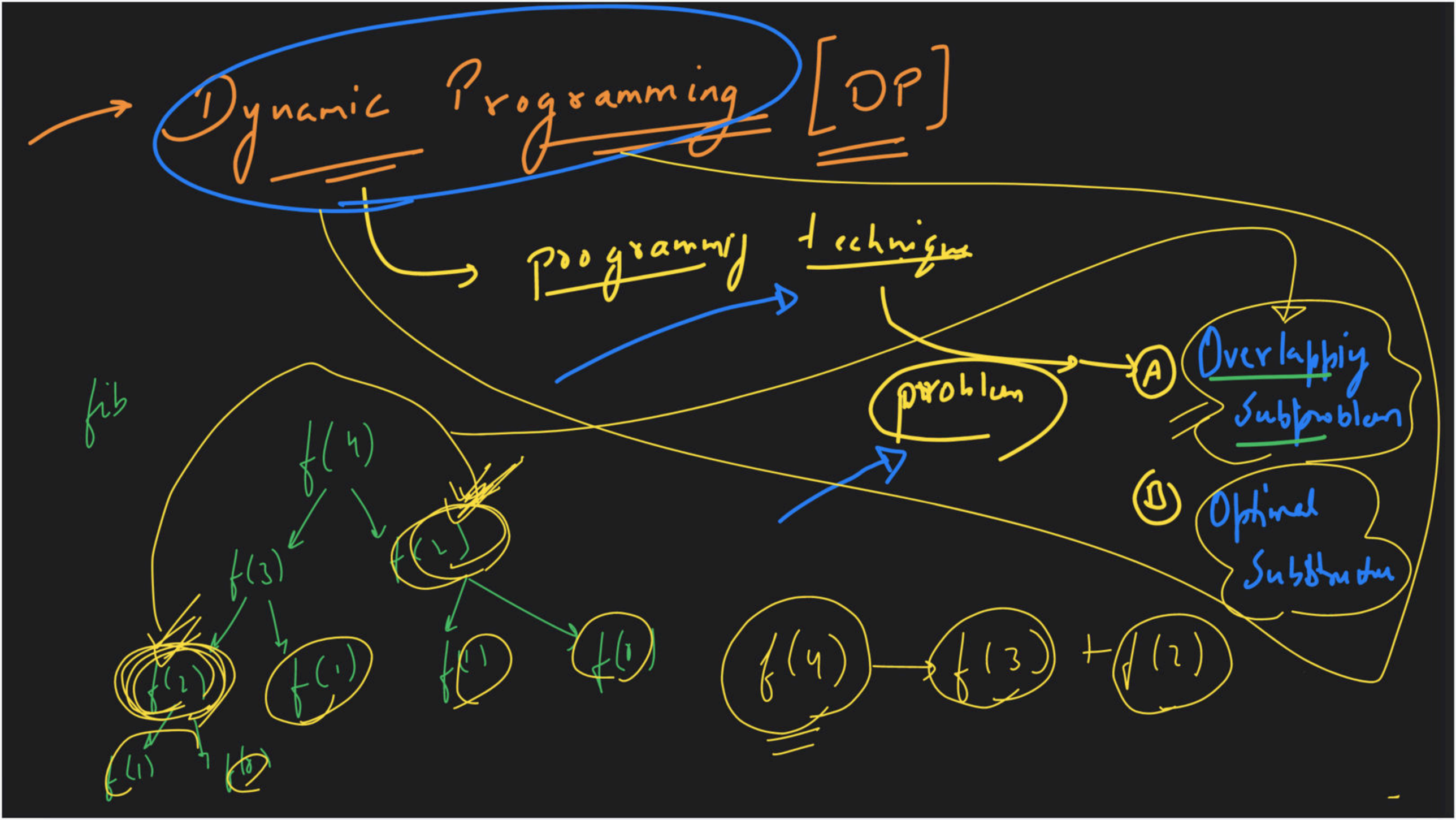
Dynamic Programming Class-1

Special class

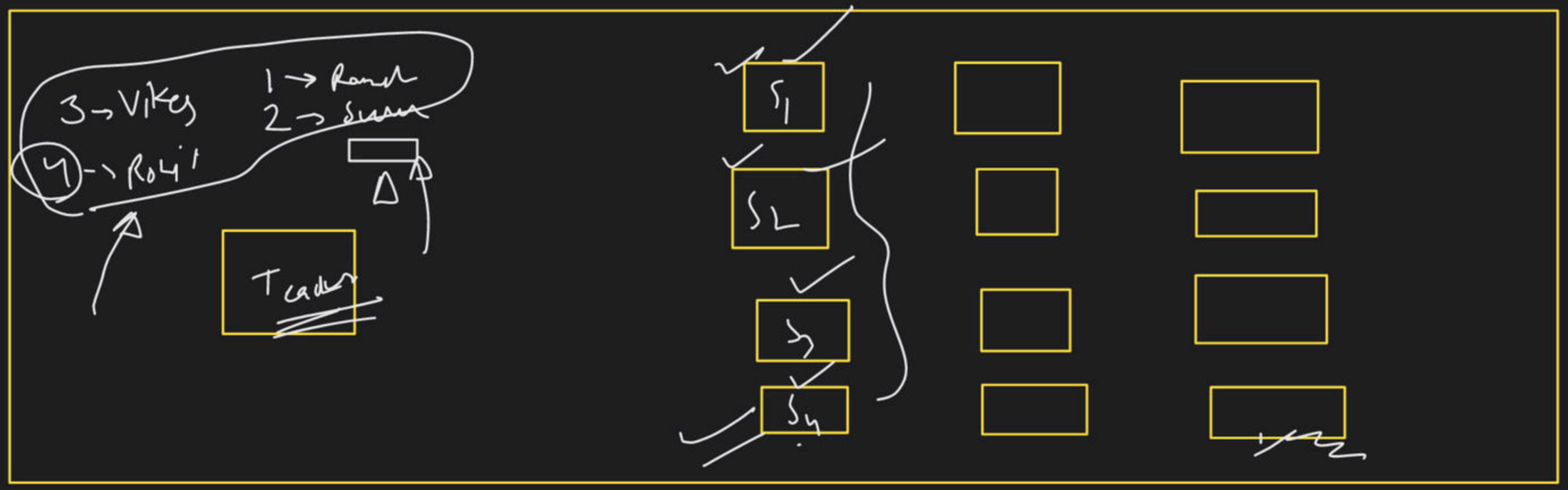
Love Babbar • Jan 5, 2024

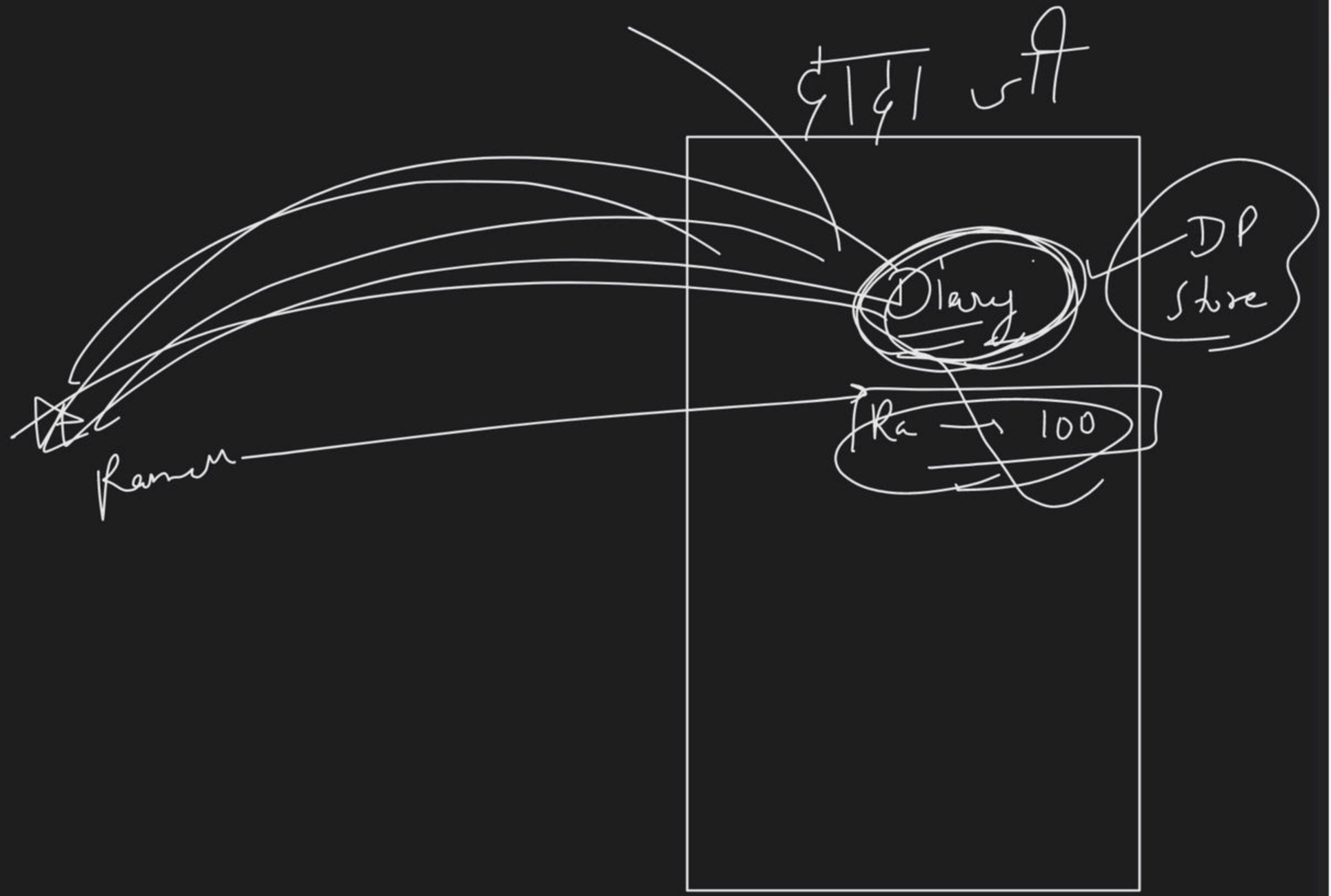

Dynamic Programming [DP]
Programming technique
problem
A
Overlapping Subproblem
B
Optimal Substructure
fib
f(4)
f(3)
f(2)
f(2)
f(1)
f(1)
f(1)
f(1)
f(0)
f(0)
f(4)
f(3)
f(2)

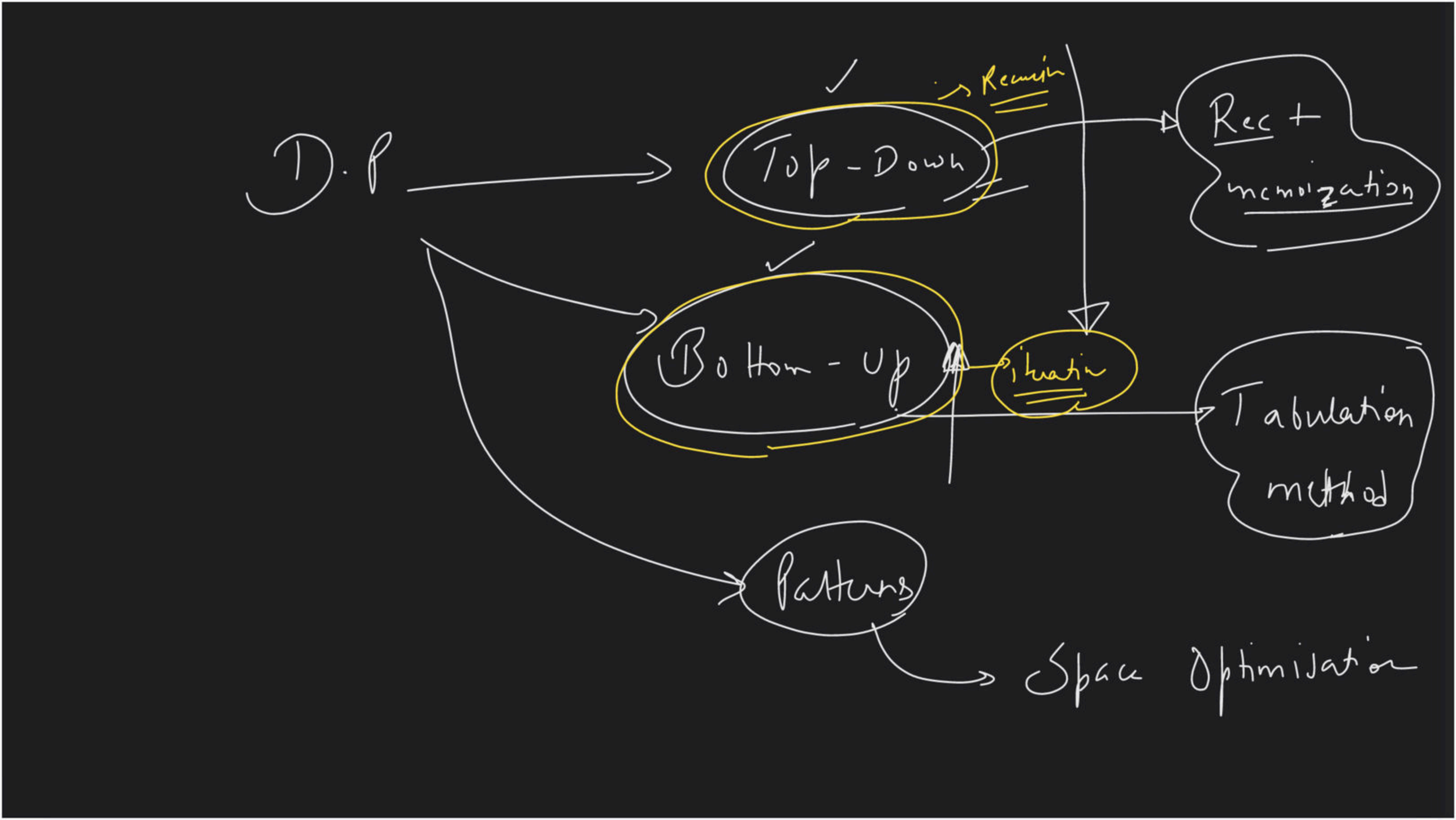
D.P → ?

→ solution

→ same problem → baar baar solve karte hai




Diary
DP
store
Ram
Ra → 100


D.P
Top-Down
Recursion
Rec + memoization
Bottom-Up
Situation
Tabulation method
Patterns
Space Optimization

Assume → Rec

→ fibonacci no:-

0, 1, 1, 2, 3, 5, 8, 13, 21 - - - - - -

2 + 3 = 5

Recursion

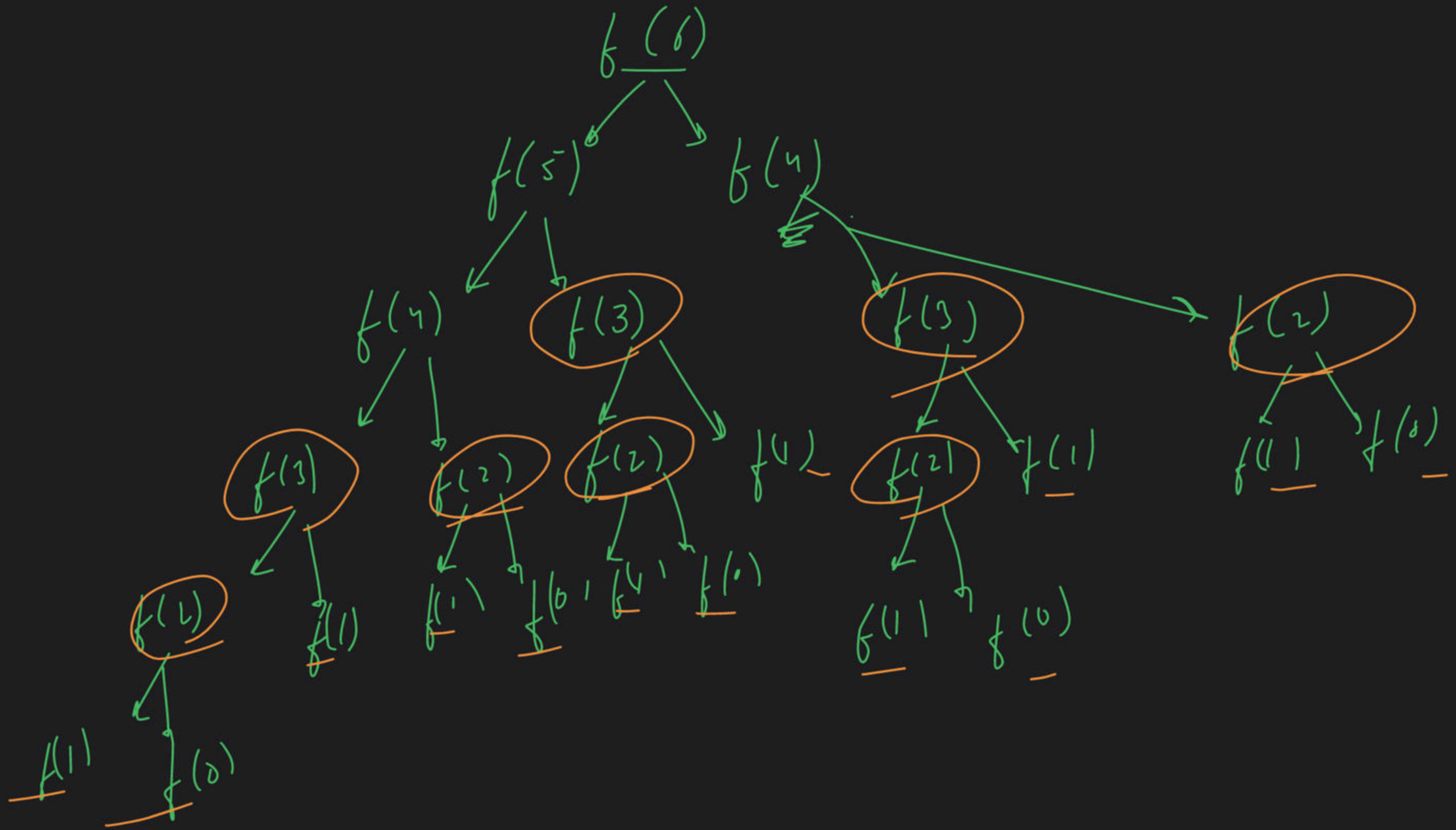
R.R → $f(n) = f(n-1) + f(n-2)$

n^{th} fibonacci no = $(n-1)^{th}$ fib no + $(n-2)^{th}$ fib no

Let $n = 6$

$f(6) = f(5) + f(4)$

6^{th} fib no = 5^{th} fib no + 4^{th} fibonacci


f(6)
f(5)
f(4)
f(4)
f(3)
f(3)
f(2)
f(3)
f(2)
f(2)
f(1)
f(2)
f(1)
f(1)
f(0)
f(1)
f(1)
f(0)
f(1)
f(0)
f(1)
f(0)
f(1)
f(0)

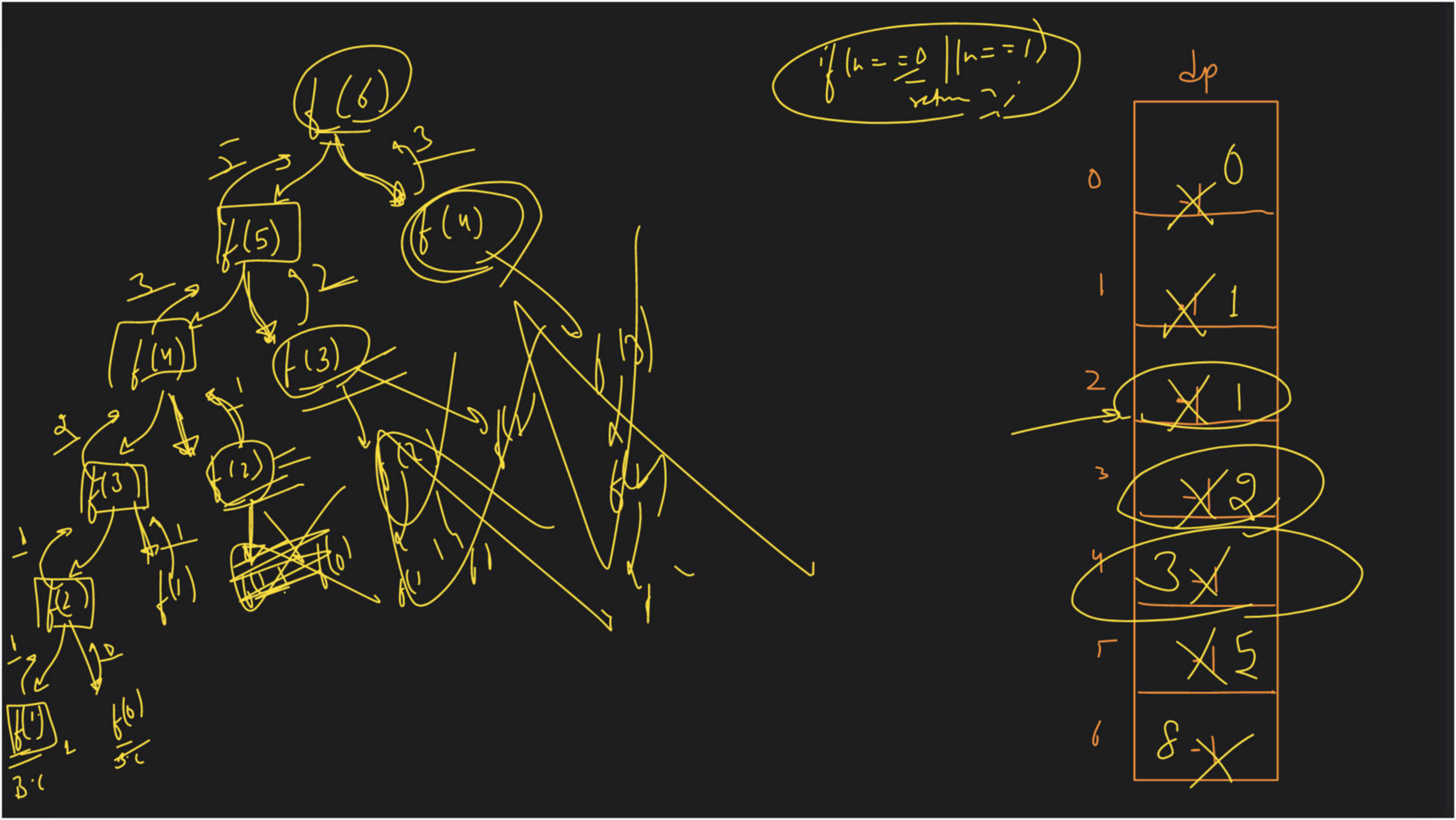
Mem

① Create a dp array

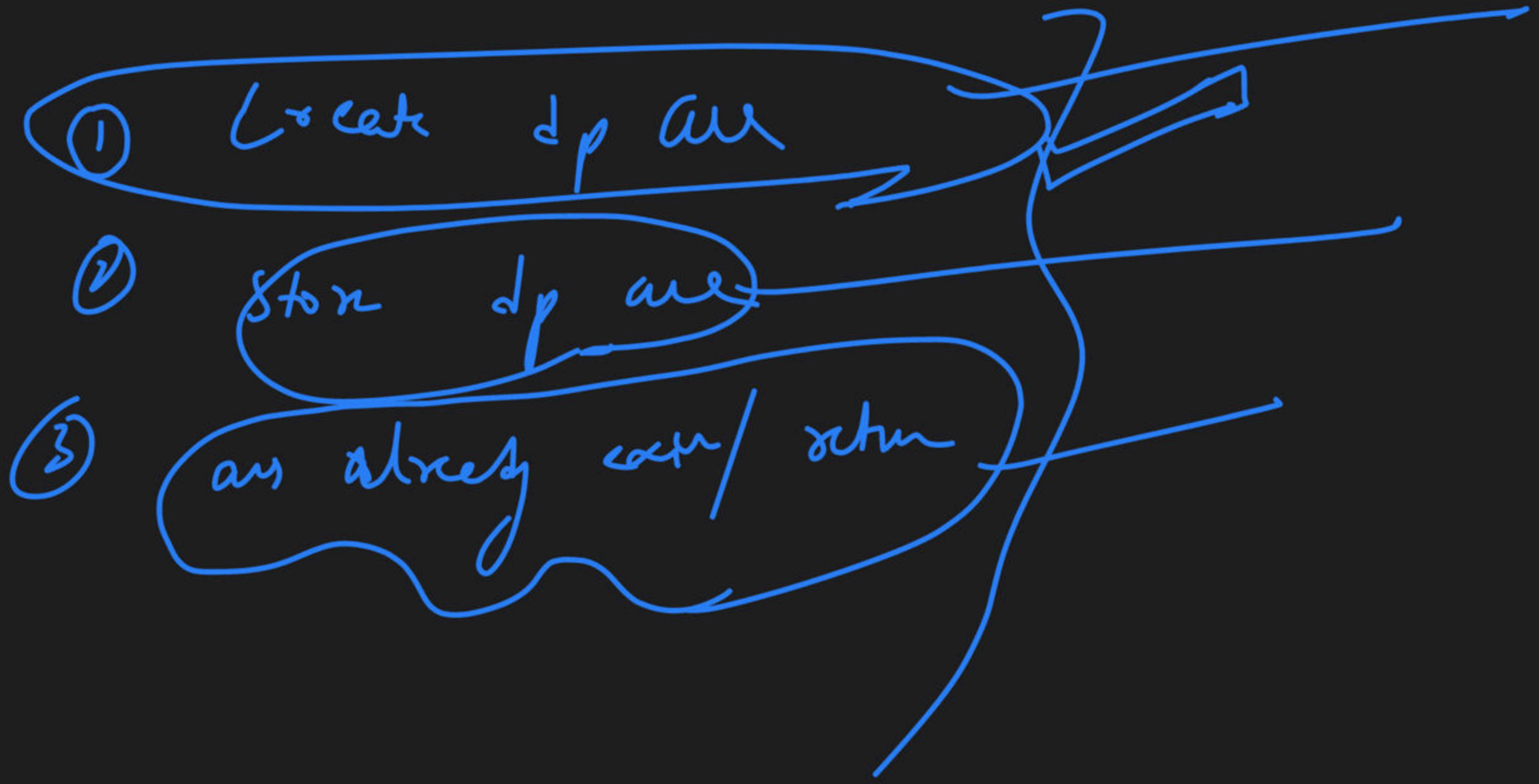
② Store/return ans in dp array

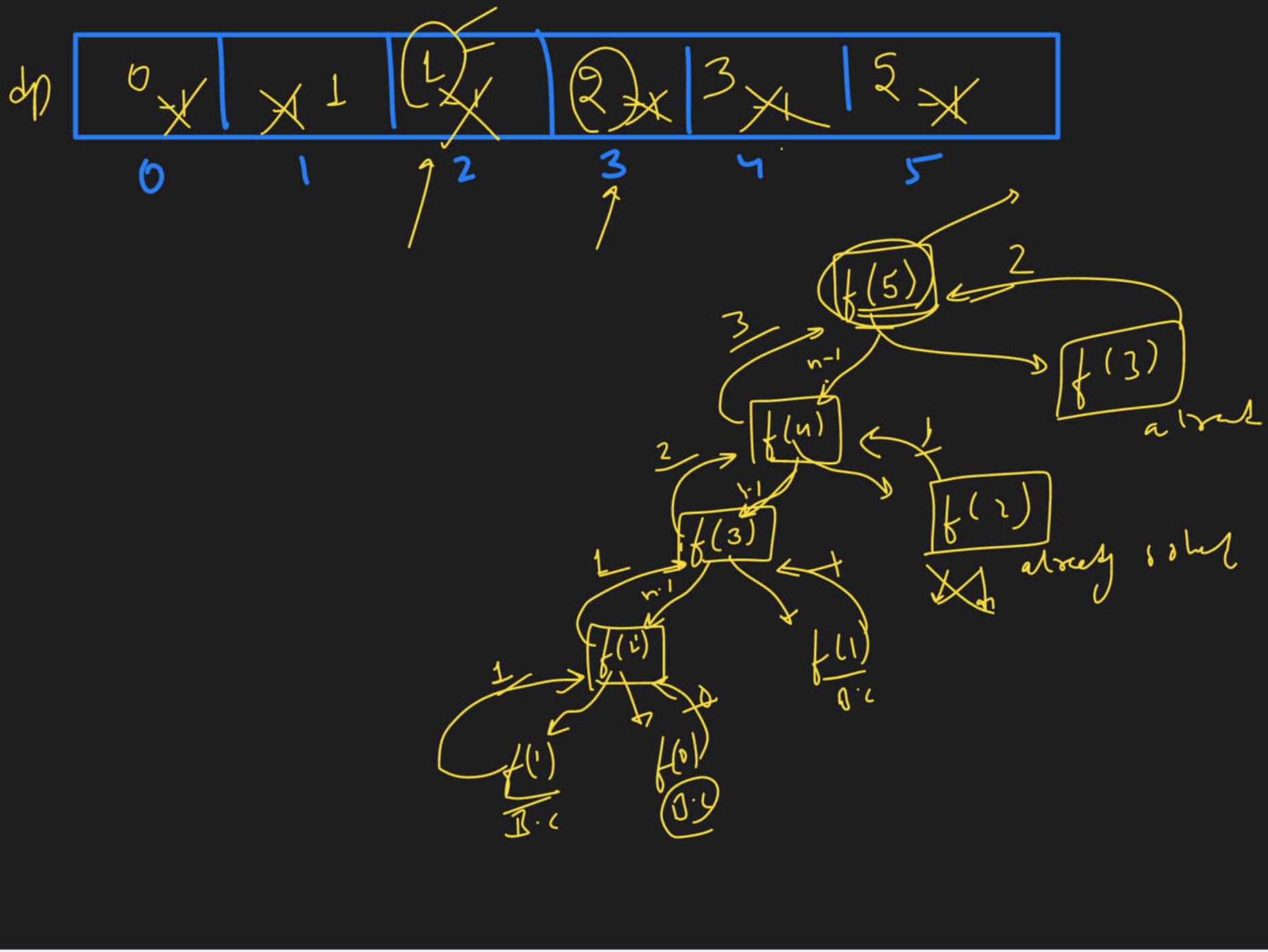
③ if ans already exists in dp array, then return

Step

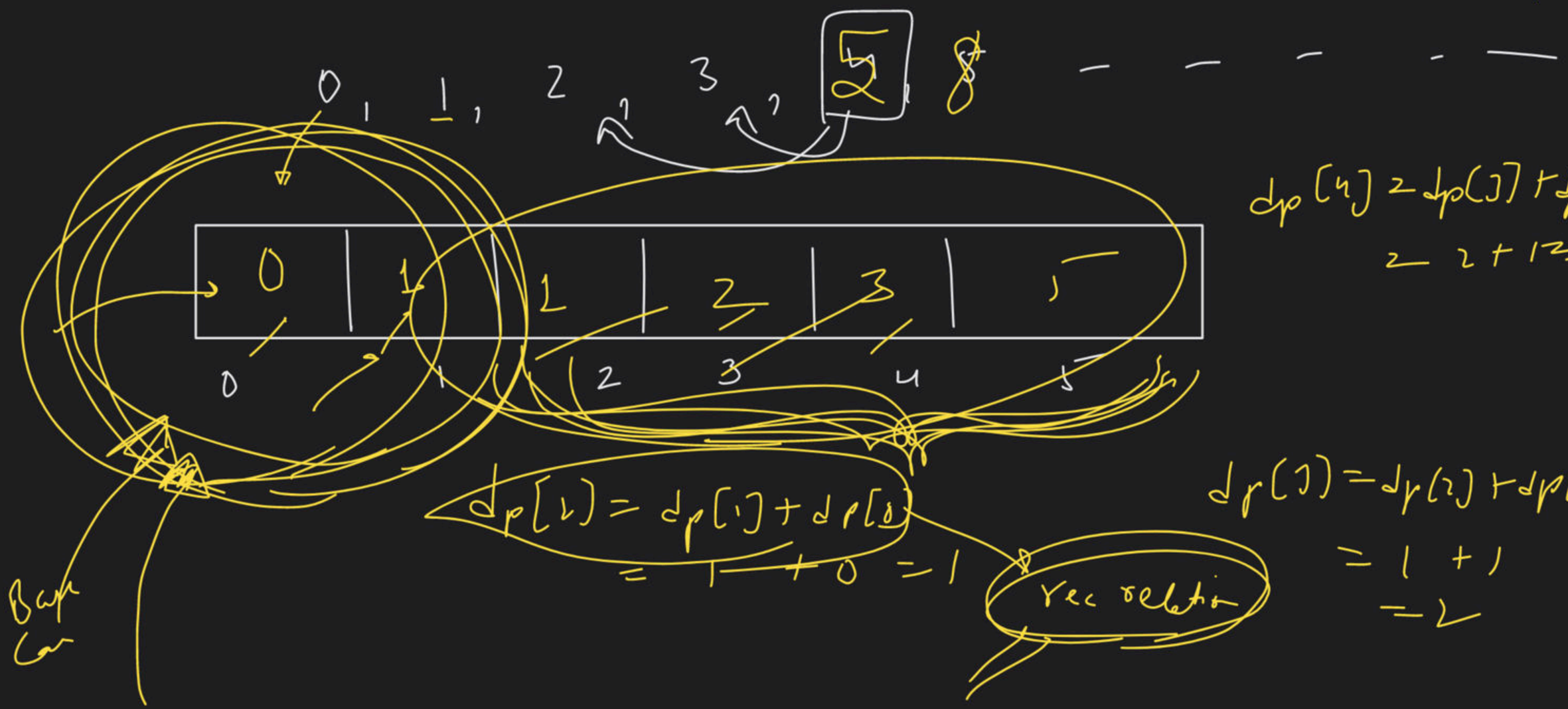

if (n == 0 || n == 1)
return n;
dp
0
1
2
3
4
5
6
f(6)
f(5)
f(4)
f(4)
f(3)
f(3)
f(2)
f(2)
f(1)
f(0)
B.C


Memoized
plain Rec
dp
0
1
2
3
4
-1


1
Locate dp arr
2
Store dp arr
3
arr already calc/ return


dp
0
1
2
3
5
0
1
2
3
4
5
f(5)
2
3
n-1
f(3)
already solved
f(4)
1
2
n-1
f(2)
already solved
f(3)
1
1
n-1
f(1)
f(1)
0.5
1
f(1)
1.5
f(0)
0.5

→ Tabulation method (Bottom-up)



Tabulation

1. → create dp array

2. → Analyse → Base case → fill dp array according

3. → fill remaining dp array using logic / formula / recursive relation

Space Optimisation :-

int solve

fib

Rec

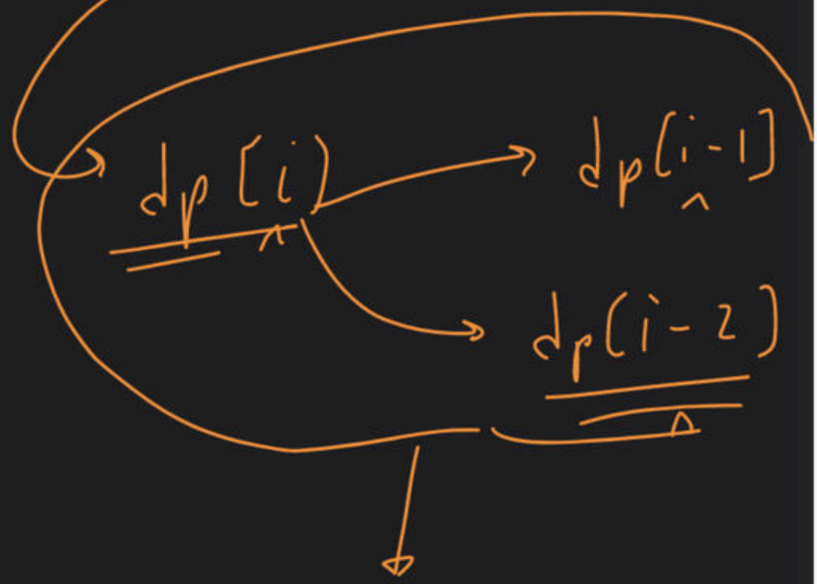
T/D / Mem

B-U / Tab

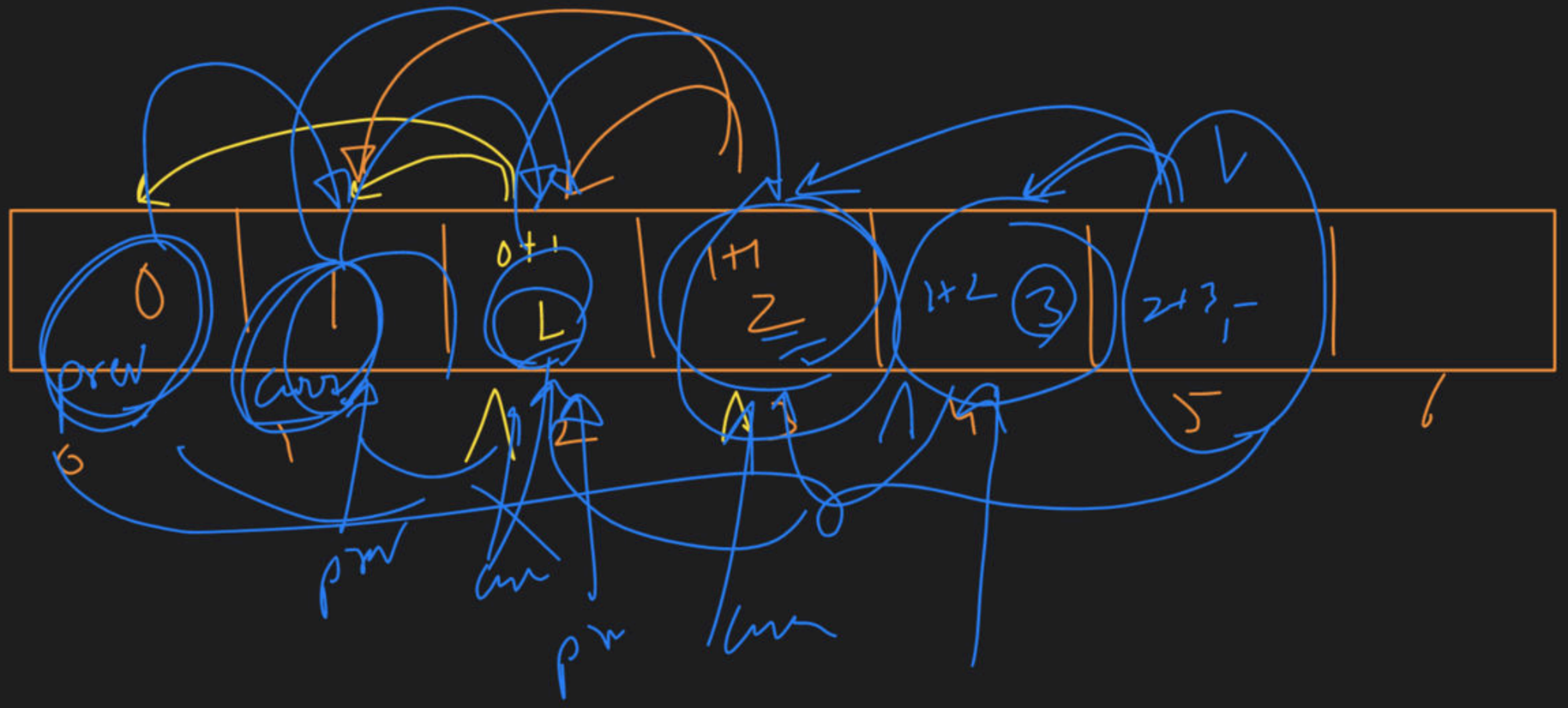
S.O

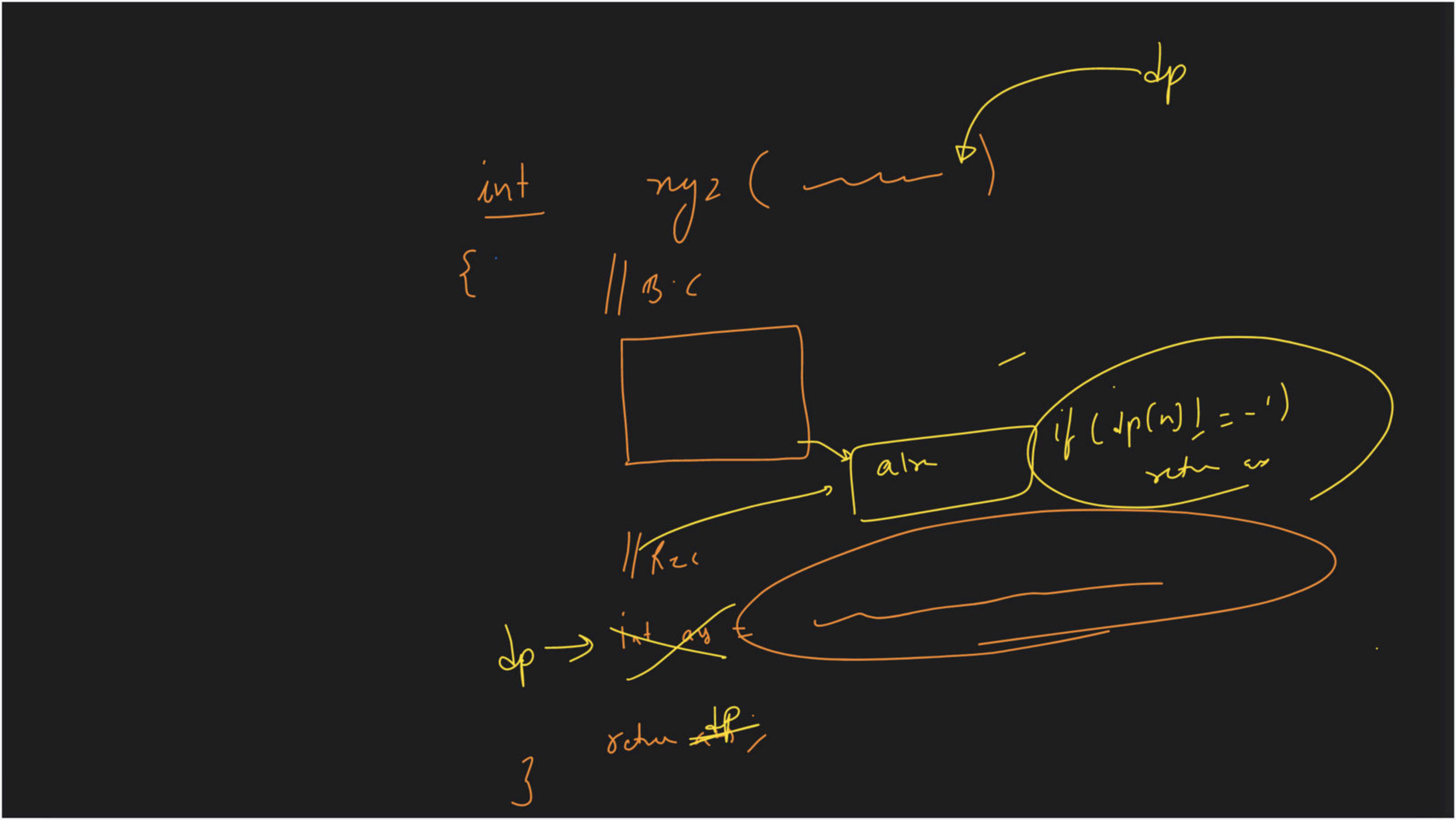
```
int solveUsingTab(int n)
{
    vector<int> dp(n+1, -1);

    dp[0] = 0;
    dp[1] = 1;

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

Space Optimization



Kya koi pattern bnaa hai




dp
int xyz ( )
{
// B.C
alx
if (dp(n) != -1)
// Rec
dp → int dy =
return dp;
}