

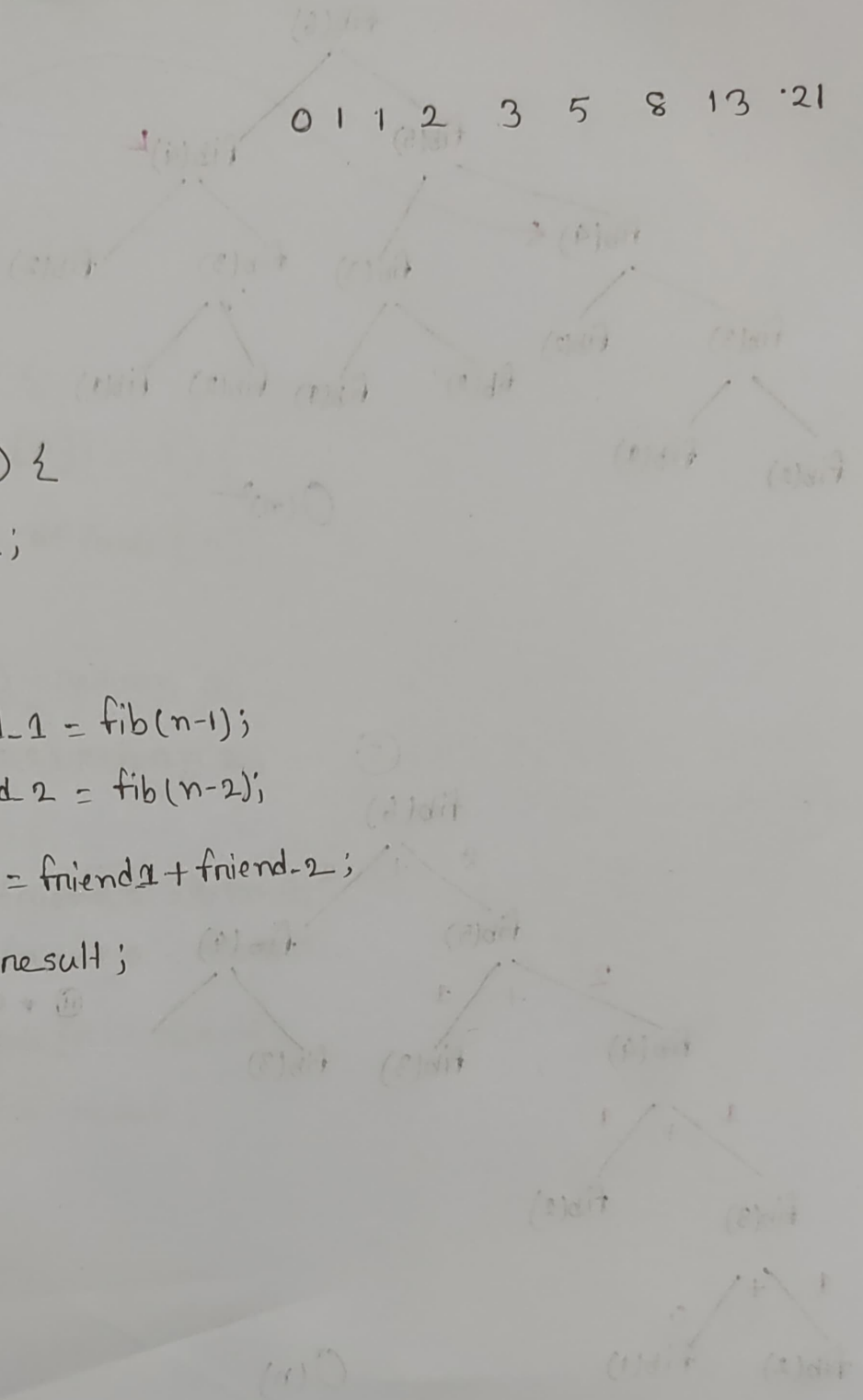
# # Dynamic Programming (Topic-4)

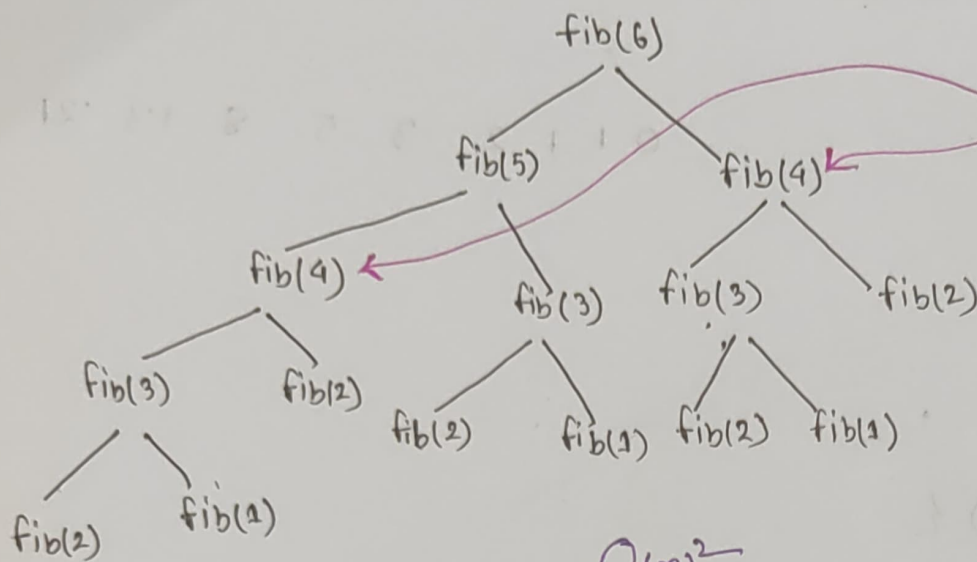
↳ Reuse, never do the same work again

lets, Fibonacci

```
int fib(int n) {  
    if (n == 1) {  
        return 0;  
    }  
    else-if (n == 2) {  
        return 1;  
    }  
    else {  
        int friend_1 = fib(n-1);  
        int friend_2 = fib(n-2);  
        int result = friend_1 + friend_2;  
        return result;  
    }  
}
```

0 1 1 2 3 5 8 13 21



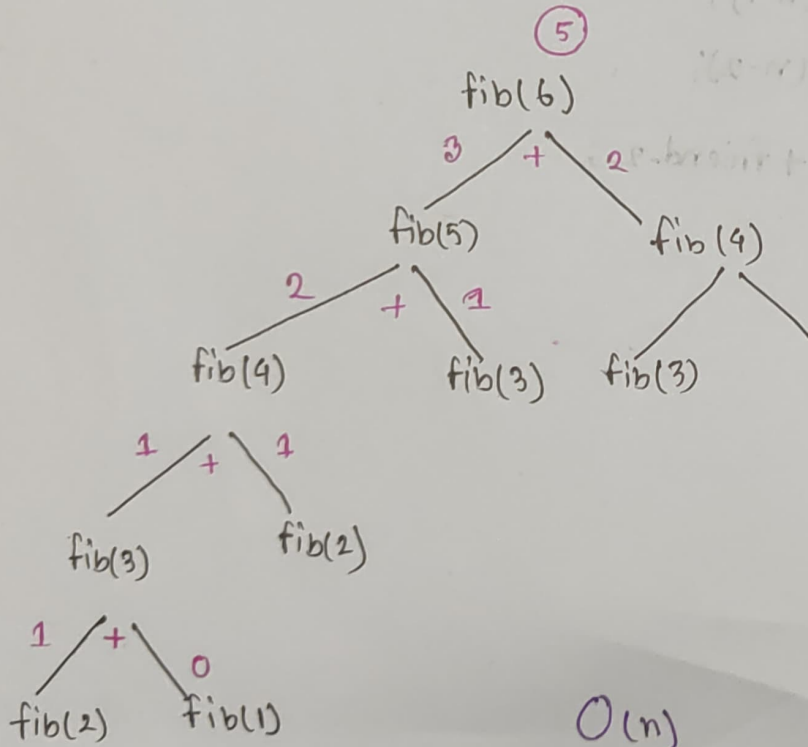


overlapping subproblem

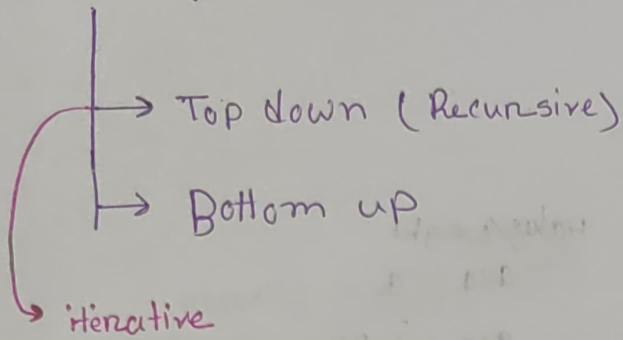
$O(n)^2$

DP Table

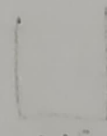
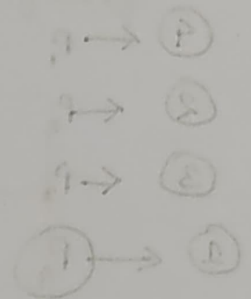
①	1	0
2	1	1
3	-1	1
4	-1	2
5	-1	3
6	-1	5



Two way of implement a DP solution



```
int fib(int n) {  
    if (dpTable[n] != -1) {  
        return dpTable[n];  
    }  
    if (n == 1) return 0;  
    elseif (n == 2) return 1;  
    else if {  
        int friend_1 = fib(n-1);  
        int friend_2 = fib(n-2);  
        dpTable[n] = result;  
        return result;  
    }  
}
```



# Optimal Substructure Property

↳ Every Possible way to solve

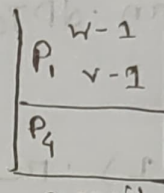
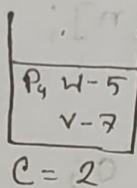
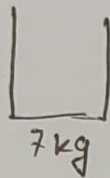
Type-2

Bottom up (tabular)

Greedy Recap -

Product	weight	value	value/weight
④ ← P <sub>1</sub>	1	1	1/1 = 1
② ← P <sub>2</sub>	3	4	4/3 = 1.33
③ ← P <sub>3</sub>	4	5	5/4 = 1.25
① ← P <sub>4</sub>	5	7	7/5 = 1.4

capacity = 7 kg



P<sub>1</sub>, P<sub>4</sub>

w → 1, 5

v → 1 + 7 = 8

but here greedy doesn't give us optimal solution.

the optimal solution is P<sub>2</sub>, P<sub>3</sub>. for achieving this

we need to use Dynamic Programming.



# DP 0/1 knapsack

P	W	value
P <sub>1</sub>	1	1
P <sub>2</sub>	3	4
P <sub>3</sub>	4	5
P <sub>4</sub>	5	7

capacity = 7 kg

capacity → base column

Weight capo	0	1	2	3	4	5	6	7
P <sub>1</sub> 1 <sub>(1)</sub>	0	1	1	1	1	1	1	1
P <sub>2</sub> 3 <sub>(4)</sub>	0	1	1	4	5	5	5	5
P <sub>3</sub> 4 <sub>(5)</sub>	0	1	1	4	5	6	6	9*
P <sub>4</sub> 5 <sub>(7)</sub>	0	1	1	4	5	6	8	9*

maximum profit gain here,

Table make Process ↓

- multiple product  
ଅଧିକ ମାତ୍ରା
- Same weight + same capacity ଏକ ସମାନ ମାତ୍ରା ଓ କ୍ୟାପାସିଟି କୋପି ହେଉଛି ।
- Same ଟାଇମ୍ ଆବଶ୍ୟକ, then, take or not take ନିର୍ଣ୍ଣୟ କରାଯାଏ, ନିଶ୍ଚିତ ନୁହେଁ, ନା ନିଶ୍ଚିତ ନୁହେଁ ।
- Table element → value
- take, not take ଏକ max value ନିର୍ଣ୍ଣୟ

P<sub>2</sub>, Capacity - 3 - 3 = 0

Take - 4 + 0 = 4

Not take - 1

iii capacity - 5 - 3 = 2

Take - 4 + 1 = 5

NTake - 1

v capacity - 7 - 3 = 4

Take - 4 + 1 = 5

NT - 1

Solution Process ↓

① max value ଡିମାଣ୍ଡ ଏକା ଓ ଡାହାଣ ପାର୍ଶ୍ୱରେ ଡିମାଣ୍ଡ ଚଳୁଥିବା...

② ଡିମାଣ୍ଡ ଡାହାଣ ପାର୍ଶ୍ୱରେ ଡିମାଣ୍ଡ ଆସୁଛି ଏକ product ଡିମାଣ୍ଡ

③ Weight ସମାନ ଓ ଡାହାଣ ପାର୍ଶ୍ୱରେ ଡିମାଣ୍ଡ jump ହେଉଛି

④ ସମସ୍ତ Base column ଓ ଆସୁଛି ଏହା work done ans,

ii capacity - 4 - 3 = 1

Take - 4 + 1 = 5

not Take - 1

iv capacity - 6 - 3 = 3

Take - 4 + 1 = 5

NT - 1

P<sub>3</sub> capacity - 4 - 4 = 0

T - 5 + 0 = 5

NT - 5

ii cap - 5 - 4 = 1

T - 5 + 1 = 6

NT - 5

ii cap - 6 - 4 = 2

T - 5 + 1 = 6

NT - 6

iv cap - 7 - 4 = 3

T - 5 + 1 = 6

NT - 5

P<sub>3</sub>, P<sub>2</sub>

W - 4 3 = 7

V - 5 4 = 9

Ans DP

P<sub>1</sub>, P<sub>4</sub>

W ≥ 1 5 = 6

V ≥ 1 7 = 8

greedy

P<sub>4</sub> cap - 5 - 5 = 0

T - 7 + 0 = 7

NT - 6

ii cap - 6 - 5 = 1

T - 7 + 1 = 8

NT - 6

iii cap - 7 - 5 = 2

T - 7 + 1 = 8

NT - 9

# Type-3 Coin change

Optimal substructure Property

Example - Amount = 11

Coin = [1, 5, 6, 8]

J →

i ↓

coin \ amount	0	1	2	3	4	5	6	7	8	9	10	11
1	0	1	2	3	4	5	6	7	8	9	10	11
5	0	1	2	3	4	1	2	3	4	5	2	3
6	0	1	2	3	4	1	1	2	3	4	2	2
8	0	1	2	3	4	1	1	2	1	2	2	2

Solving approach →

① wheather we'll take it or not

$$\text{amount} = 5 - 5 = 0$$

take :  $1 + 0[\text{index's value}]$

Not take : 5

② take only the minimum value

$$\text{amount} = 9 - 5 = 4$$

$$T = 1 + 4 = 5$$

$$NT = 9$$

$$\text{amount} = 10 - 5 = 5$$

$$T = 1 + 1 = 2$$

$$NT = 10$$

$$\text{amount} = 11 - 5 = 6$$

$$T = 1 + 2$$

$$NT = 11$$

$$\text{amount} = 6 - 6 = 0$$

$$T = 1 + 0 = 1$$

$$NT = 6$$

$$\text{amount} = 8 - 8 = 0$$

$$T = 1$$

$$NT = 3$$

$$\text{amount} = 10 - 6 = 4$$

$$T = 1 + 4 = 5$$

$$NT = 2$$

$$\text{amount} = 11 - 6 = 5$$

$$T = 1 + 1 = 2$$

$$NT = 3$$

Solve =

$$6 + 5 = 11$$

$$\text{amount} = 6 - 5 = 1$$

$$\text{Take} = 1 + 1 = 2$$

$$\text{Not take} = 6$$

$$\text{amount} = 7 - 5 = 2$$

$$\text{Take} : 1 + 3$$

$$\text{not Take} = 7$$

$$\text{amount} = 8 - 5 = 3$$

$$T = 1 + 3 = 4$$

$$NT = 8$$

# Example - 2

amount = 6

coin = [2, 3, 5]

coin ↓ amount →	0	1	2	3	4	5	6
2	0	∞	1	∞	2	∞	3
3	0	∞	1	1	2	2	2
5	0	∞	1	1	2	1	2

amount - 3, 5 - 3 = 2

T - 1 + 1 = 2

NT - ∞

6 - 3 = 3

T - 1 + 1 = 2

NT - 3

6 - 5 = 1

T - 1 + ∞ = ∞

NT - 2

Ans 3, 3

3 + 3 = 6