Q1 Given integer N. Return minimum count of numbers, sum of whose squares is equal to N.

Ex1 $N = 100 \Rightarrow 10^2$ ans = 1

Ex2 $N = 101 \Rightarrow 10^2 + 1^2$ ans = 2

Ex3 $N = 13 \Rightarrow 3^2 + 2^2$ ans = 2

Ex4 $N = 6 \Rightarrow 2^2 + 1^2 + 1^2$ ans = 3

Did you think greedily ?

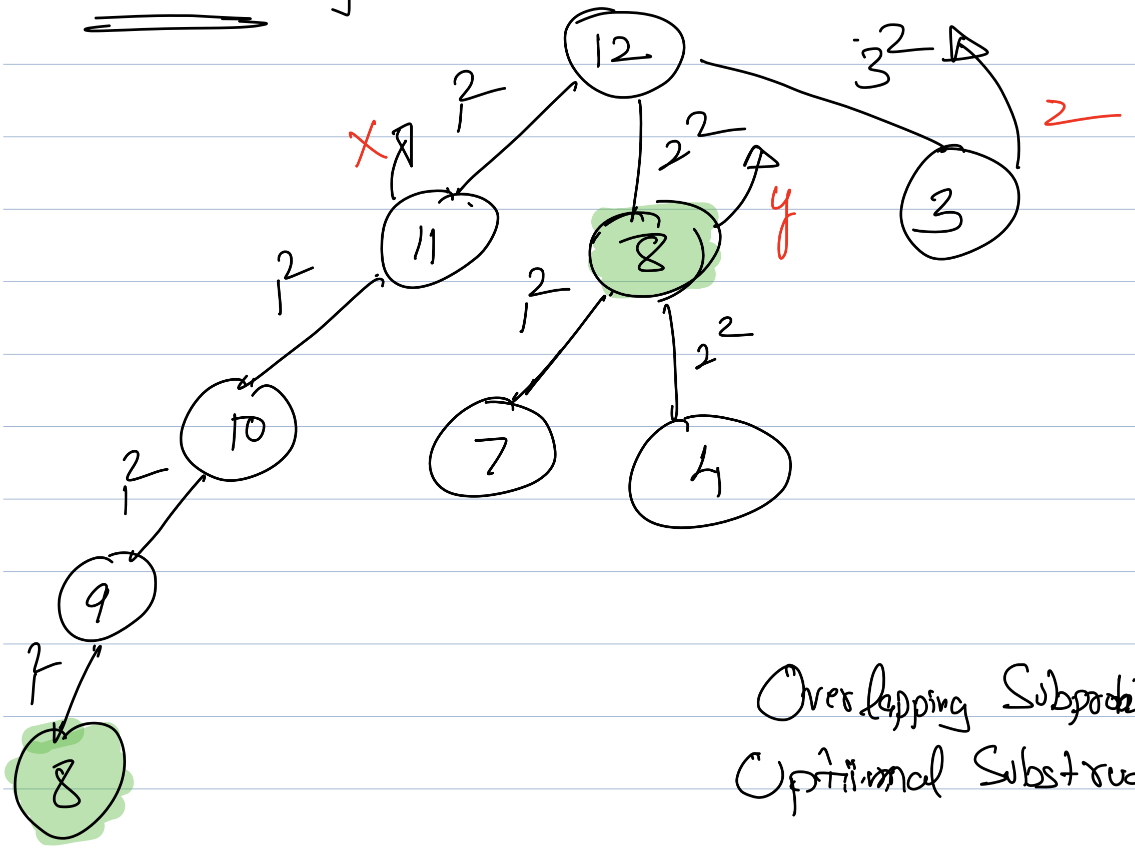
Ex : $N = 12 \Rightarrow 3^2 + 1^2 + 1^2 + 1^2$

~~ans = 4~~

$\Rightarrow 2^2 + 2^2 + 2^2$

ans = 3

Backtracking



Overlapping Subproblems ✓

Optimal Substructure ✓

1) Element of choice.

$N \Rightarrow 12 \quad [1, .\, \text{floor}(\sqrt{n})]$

$\hookrightarrow [1, 3]$

2) What does a state represent.

$dp[i] \Rightarrow X$

$X \Rightarrow$ minimum count of numbers, sum of whose squares is equal to i

3) Recurrence relation

$$dp[n] \Rightarrow 1 + \mathop{I}_{i=1}^{i=\text{floor}(\sqrt{n})} \min(n-i^2)$$

Top down

```
dp [n+1] = {-1} , dp[0] = 0            Store bse cases.

int count (int n) {                    Checking if previously calculated

  if ( dp(n] != -1)                    calculating ans
                                       store
         return dp (n]                 return the ans

    int ans = N

 for ( int i = 1 ; i*i <= N ; i++ ) {

      ans = min (ans, 1 + count (N-i^2));

 }

    dp [n] = ans.                      TC: O(n√n)

   return ans;                         SC : O(n)

}
```

Q2

N houses in a straight line
You are a thief.

Each house has some gold in it.
Each house also has alarm system in it.

If you rob the house i, then the houses i-1 and i+1 would be notified and you cannot rob them

Maximize the amount of gold that you can collect.

Ex1 A :

0	1	2	3
10	50	20	60

⇒ 110

Ex2 A :

0	1	2	3
10	20	10	20

⇒ 40

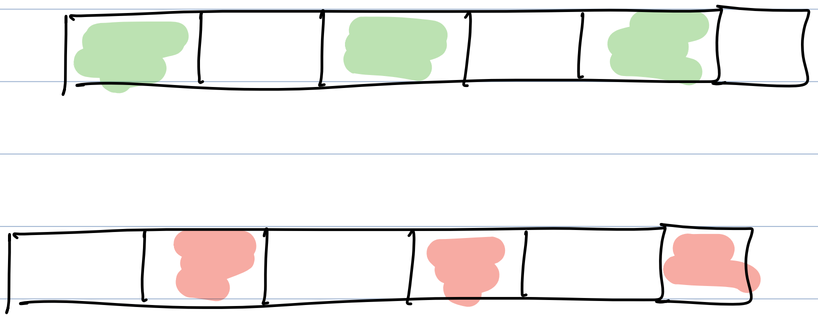
Can we go greedy

Ex

50	60	40

⇒ 2

Ex



100	1	1	100

⇒ 200

Find sum of all subset.

Find max , Find if valid

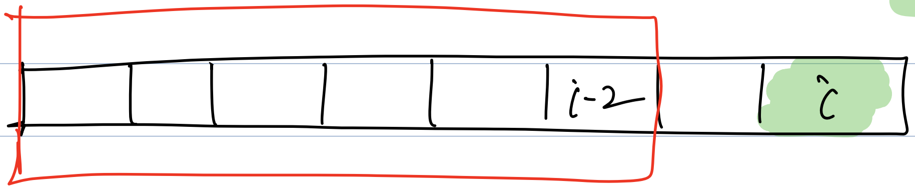
Tc : $O(N \times 2^n)$

dp[i]

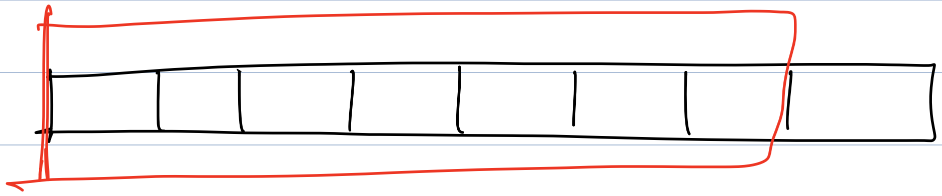
1) Element of choice

→ max gold robbed including ith house.

→ max gold robbed till ith house



$A[i] + dp[i-2]$ If robbed.



$dp[i-1]$ If not robbed,

2) State

$dp[i]$ → maximum answer till i^{th} house.

3) Recursive relation

$$dp[i] = \max(A[i] + dp[i-2], dp[i-1])$$

4) answer = $dp[N]$

Top down

```
dp [n] = {-1}
dp [0] => arr [0];
dp [1] => max (arr [0], arr [1])
int   max-gold ( int n , int arr [ ] )
    if ( dp [n] != -1 )
        return dp [n]

    dp [n] =   max ( arr [n] + max-gold (n-2)
                                 max-gold (n-1) );

    return dp [n];
}
```

TC : O(n)

SC : O(n)

LIS { Longest Increasing Subsequence }.

Given an array. Find it's LIS.

Ex1 A :

0	1	2	3	4	5	6
2	3	5	1	10	3	7

ans = 4

7

0	1	2	3	4	5	6
2	3	5	1	10	3	7

1) Choices if i is included ⇒ All element

< arr[i].

2) State:

dp(i) ⇒ max length till i.

max length including i.

3) Recurrence relation

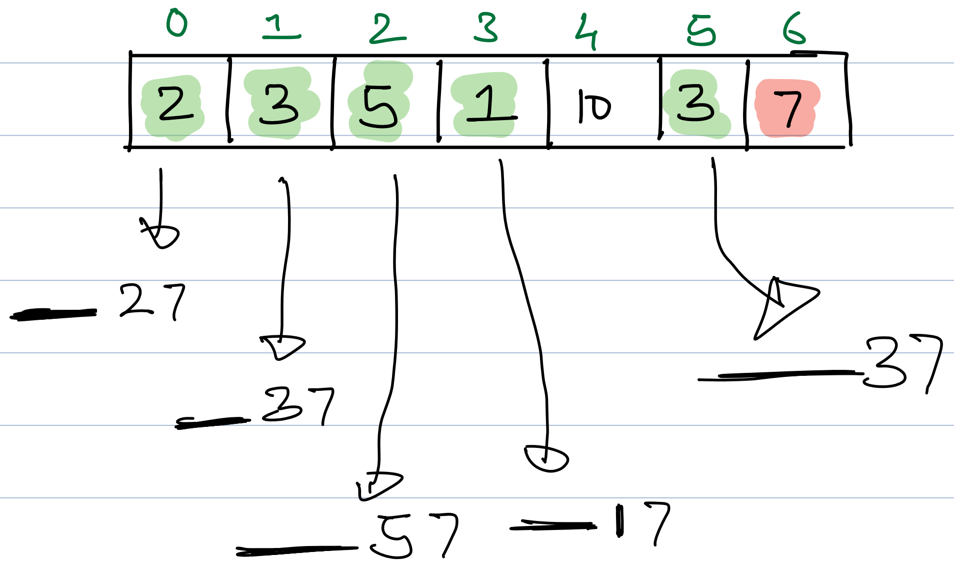
$$dp[i] \Rightarrow \Big|_{j=0}^{i-1} \max(dp[j]) + 1$$

$$arr[j] < arr[i]$$

4) What is ans

$$\Rightarrow dp[n-1]$$

$$\max(dp)$$



0	1	2	3	4	5	6
2	3	5	1	10	3	7

0	1	2	3	4	5	6	7
2	3	5	1	10	3	7	1

0	1	2	3	4	5	6	7
1	2	3	1	4	2	4	1
-1	(0)	1	-1	2	3	2	-1

⇒

Total states ⇒ n.

Total work at each state ⇒ n.

TC : $O(n^2)$

SC : $O(n)$

Same for top down & bottom up

→ 2 3 0 → 3 5 0

0	1	2	3	4	5	6	7
2	3	5	1	10	3	7	1

0	1	2	3	4	5	6	7
1	2	3	1	4	2	4	1