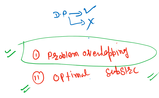
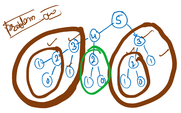
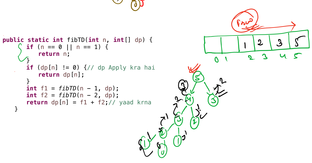
Lec-54

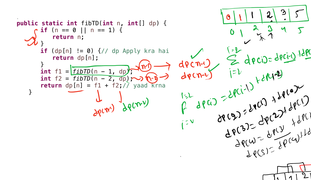




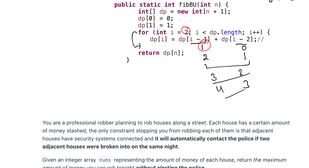
```
public static int fib(int n) {
    if (n == 0 || n == 1) {
        return n;
    }
    int f1 = fib(n - 1);
    int f2 = fib(n - 2);
    return f1 + f2;
}
```

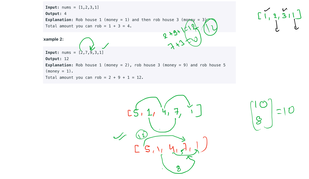



```
public static int fibTD(int n, int[] dp) {
    if (n == 0 || n == 1) {
        return n;
    }
    if (dp[n] != 0) {// dp Apply kra hai
        return dp[n];
    }
    int f1 = fibTD(n - 1, dp);
    int f2 = fibTD(n - 2, dp);
    return dp[n] = f1 + f2;// yaad krna
}
```

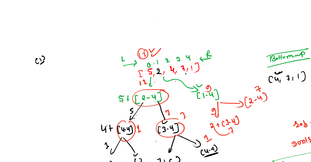



```
public static int fibTD(int n, int[] dp) {
    if (n == 0 || n == 1) {
        return n;
    }
    if (dp[n] != 0) {// dp Apply kra hai
        return dp[n];
    }
    int f1 = fibTD(n - 1, dp);
    int f2 = fibTD(n - 2, dp);
    return dp[n] = f1 + f2;// yaad krna
}
```






```
public static int fibBU(int n) {
    int[] dp = new int[n + 1];
    dp[0] = 0;
    dp[1] = 1;
    for (int i = 2; i < dp.length; i++) {
        dp[i] = dp[i - 1] + dp[i - 2];//
    }
    return dp[n];
}
```



You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security systems connected and **it will automatically contact the police if two adjacent houses were broken into on the same night**.

Given an integer array `nums` representing the amount of money of each house, return the maximum amount of money you can rob tonight **without alerting the police**.

```
Input: nums = [1,2,3,1]
Output: 4
Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3).
Total amount you can rob = 1 + 3 = 4.
```

sample 2:

```
Input: nums = [2,7,9,3,1]
Output: 12
Explanation: Rob house 1 (money = 2), rob house 3 (money = 9) and rob house 5 (money = 1).
Total amount you can rob = 2 + 9 + 1 = 12.
```

cost = [1,100,1,1,1,100,1,1,100,1]

Given an integer array `nums`, return the *length of the longest* ***strictly increasing subsequence***.

Example 1:

```
Input: nums = [10,9,2,5,3,7,101,18]
Output: 4
Explanation: The longest increasing subsequence is [2,3,7,101], therefore the length is 4.
```