Space Optimization

① What other states does our current state depend on ?

② Do we need the answers of all states to calculate the answer for current state

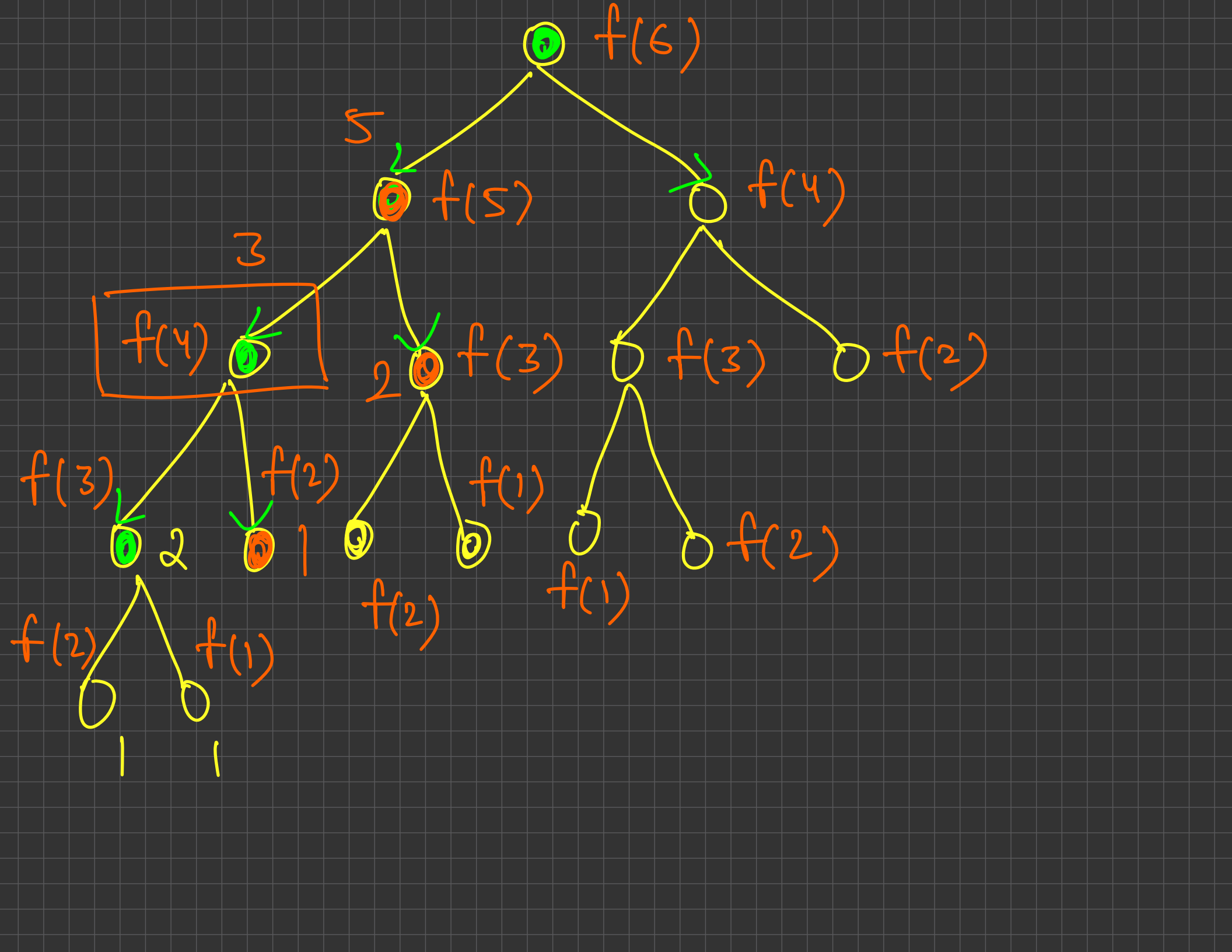
$$dp(n) = \underbrace{dp(n-1)}_{x} + \underbrace{dp[n-2]}_{y}$$

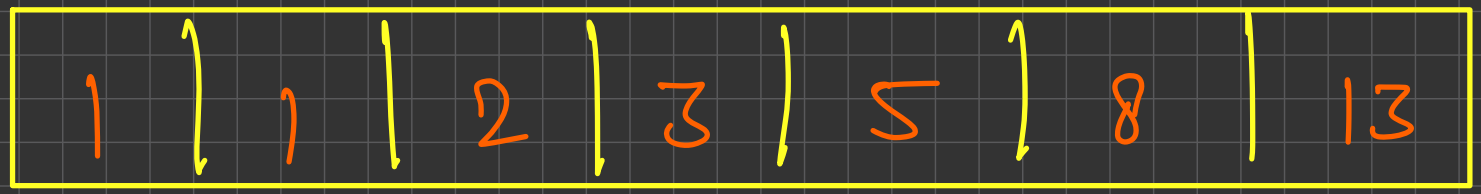
L.H.S

(R.H.S)

$dp(k)$

$dp(k-1)$ $\quad$ $dp[k-2]$


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



f(1) f(2) f(3) f(4) f(5) f(6) f(7)

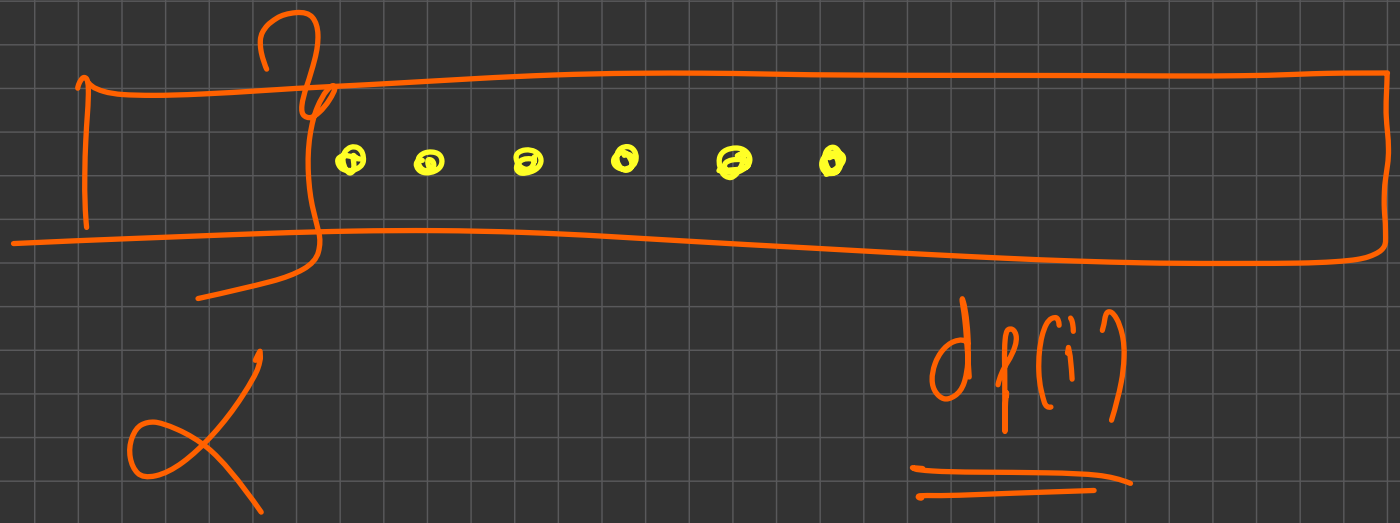
dp(n)

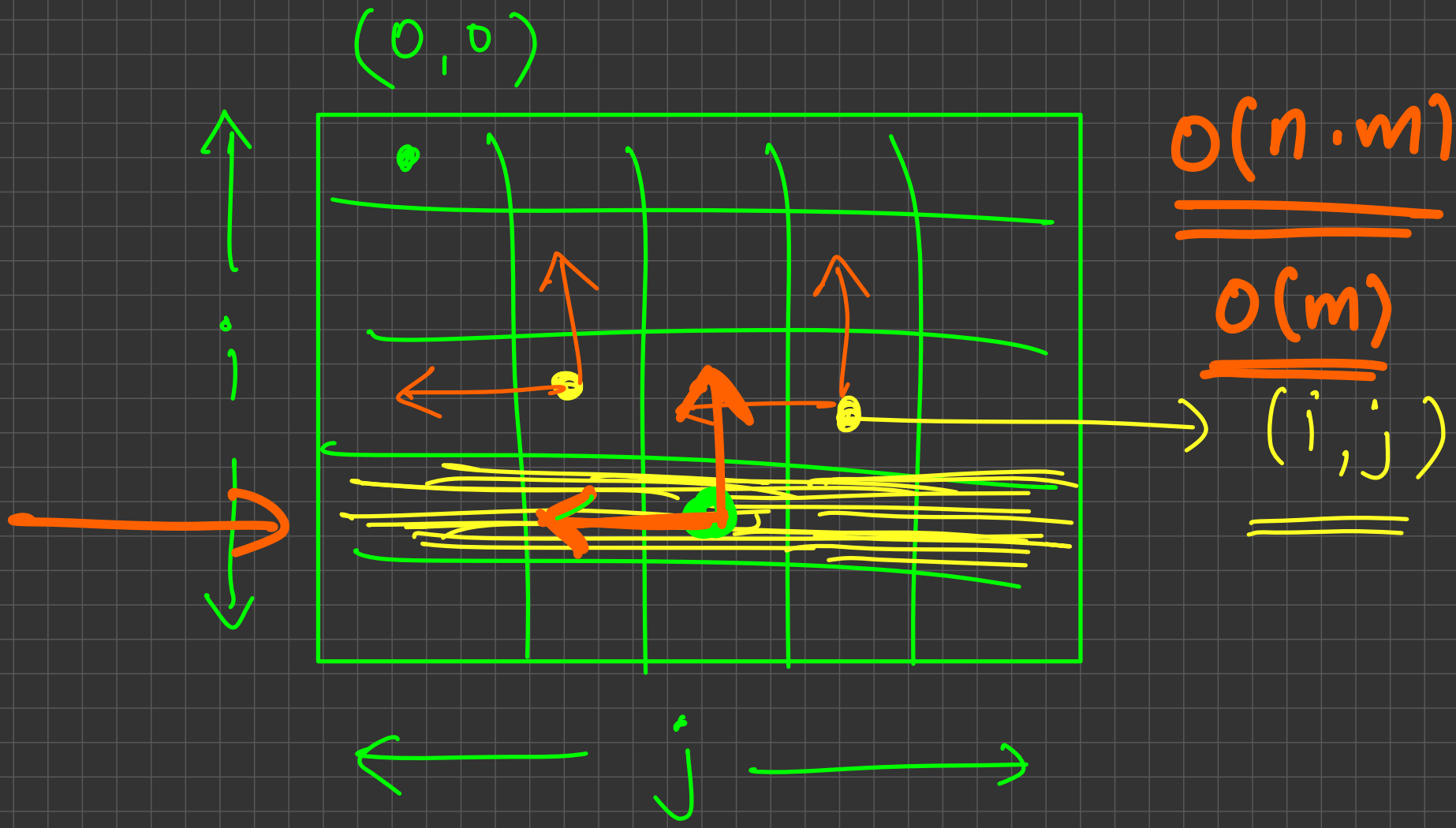
Examples $(1 \le i \le n,\ 1 \le j \le m)$

① $dp[i] = dp[i-1] + dp[i-2]$ ✓ fibo

② $dp[i] = \sum_{j=1}^{6} dp[i-j]$ dice combinations

③ $dp[i][j] = \min \begin{cases} dp[i-1][j] \\ dp[i][j-1] \end{cases}$ ✓


dp(i)


(0,0)
i
j
O(n·m)
O(m)
(i,j)

fibonacci problem ①

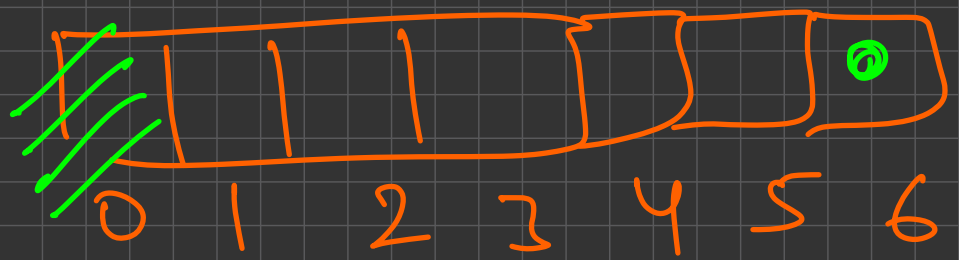
```cpp
void solve(){
    int n;
    cin >> n;
    int prev1 = 1, prev2 = 1;
    for(int i = 3; i <= n; i++){
        // finding dp[i] or ith fibonacci number here
        int current = prev1 + prev2;

        // prev1 = dp[i - 1], prev2 = dp[i - 2]
        // after ith iteration:
            // prev1 should be dp[i]
            // prev2 should be dp[i - 1]

        prev2 = prev1; // now prev2 = dp[i - 1]
        prev1 = current; // now prev1 = dp[i]
    }
    //finally prev1 = dp[n]
    cout << prev1 << endl;
}
```

dp[1], dp[2]

Div combinations ②

$$dp(i) = \sum_{j=1}^{6} dp(i-j)$$

```cpp
void solve(){
    int n;
    cin >> n;
    vector<int> prev = {1}; // only contains dp[0]
    for(int i = 1; i <= n; i++){
        // finding dp[i] or number of ways to make a sum of i
        // dp[i] = summation of all the previous states
        int current = 0;
        for(int j : prev){
            current = (current + j) % MOD;
        }
        prev.push_back(current);
        // if previous contains more than 6 elements
        // discard the first element
        if(prev.size() > 6){
            prev.erase(prev.begin());
        }
    }
    //finally the last element of prev would be dp[n]
    cout << prev.back() << endl;
}
```



Coin Combinations 2

(3)

Let's code it !