Dynamic Programming

Video - 76

Note :- This playlist is only for explanation of Qns & solutions.

See my "DP Concepts & Qns" playlist for understanding DP from scratch...

Facebook
Instagram] → codestorywithMIK

Twitter → cswithMIK

→ codestorywithMIK

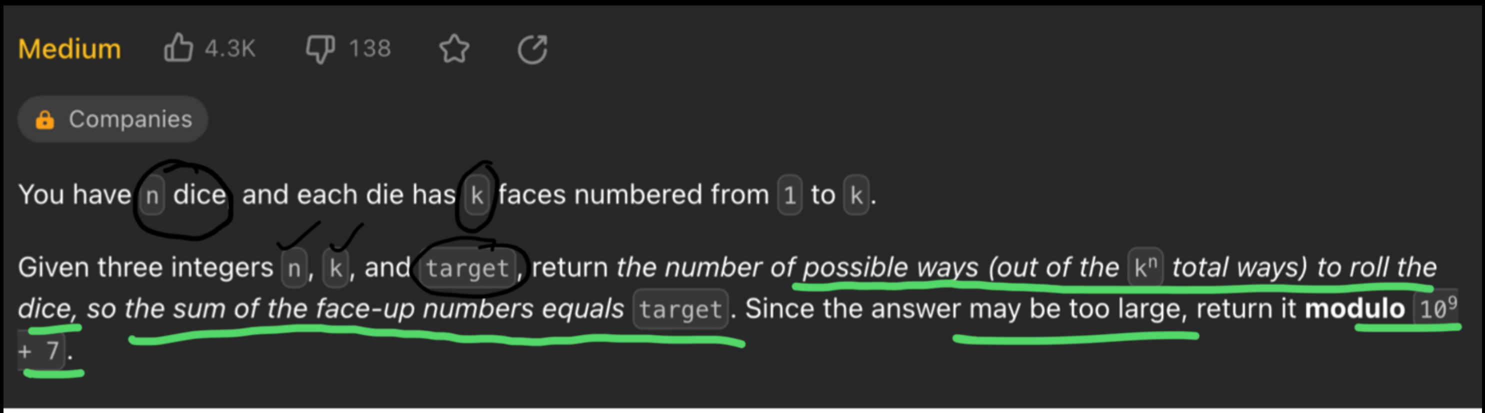
Leetcode
-1155
~~Medium~~

amazon

Microsoft

1155. Number of Dice Rolls With Target Sum

Hint



You have n dice and each die has k faces numbered from 1 to k.

Given three integers n, k, and target, return *the number of possible ways (out of the* k^n *total ways) to roll the dice, so the sum of the face-up numbers equals* target. Since the answer may be too large, return it **modulo** $10^9 + 7$.

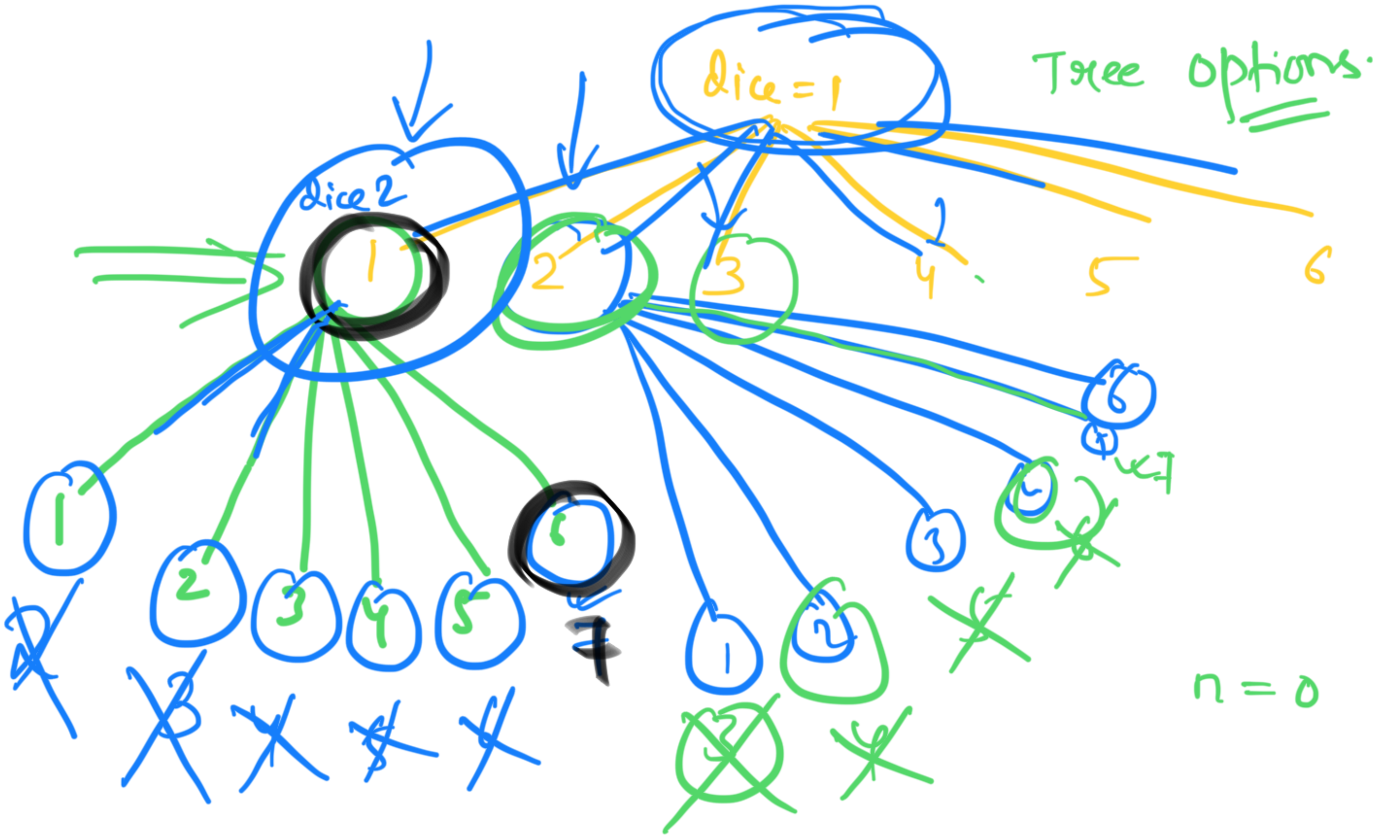
Example :- n = 2, K = 6, target = 7

Output : 6 ✓

dice 1	dice 2	target = 7
1	6	✓
2	5	✓
3	4	✓
4	3	✓
5	2	✓
6	1	✓

Thought Process

n = 2,   K = 6,   target = 7



```
Solve( int n, int K, int target) {
    if (target <0) return 0;
    if (n==0) return (target ==0);
    if ( t[n][tar] != -1) ret
int ways=0;
for (face=1 ; face<=K ; face++) {
        ways += Solve(n-1, K, target-face)
```

% MOD

}

return $t[n][target] = ways;$

}

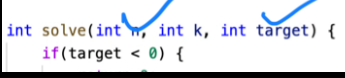
$$T.C = K * K * K \dots \text{ n dice.}$$

$$= O(K^n) \text{ possibilities.}$$

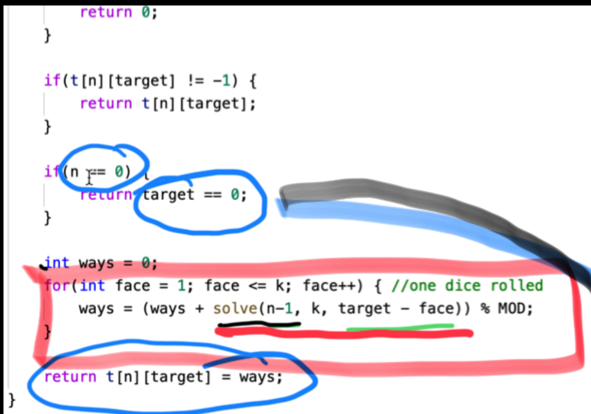
Memoize = $t[31][1001]$

Bottom UP :-

$t[n+1][target+1]$;



```
int solve(int n, int k, int target) {
    if(target < 0) {
```



```
        return 0;
    }

    if(t[n][target] != -1) {
        return t[n][target];
    }

    if(n == 0) {
        return target == 0;
    }

    int ways = 0;
    for(int face = 1; face <= k; face++) { //one dice rolled
        ways = (ways + solve(n-1, k, target - face)) % MOD;
    }

    return t[n][target] = ways;
}
```

t[i][j] = no of ways to obtain sum = "j" if we have i dices.

$\boxed{t[0][0] = 1 \;;}$

	0	1	2	target
0	1	0	0	0 ..
1	0			
2	0			
⋮	0			
n	0			

retn t[n][target]

```
for(i=1;   i<n+1 ;   i++) {

    for(j=1 ; j<target+1 ; j++) {

        // dice = i
```

```
       // target = j
        int ways = 0;
       for ( face = 1;  face <= K;  face++){

                ways=(ways+ t[i-1][j-face])%M

       }

       t[i][j] = ways;

   }

 }

 re    t[n][target];
```