DP-6

Assignment Questions

Ques1: 887. Super Egg Drop

You are given k identical eggs and you have access to a building with n floors labeled from 1 to n.
You know that there exists a floor f where 0 <= f <= n such that any egg dropped at a floor higher than f will break, and any egg dropped at or below floor f will not break.

Each move, you may take an unbroken egg and drop it from any floor x (where 1 <= x <= n). If the egg breaks, you can no longer use it. However, if the egg does not break, you may reuse it in future moves.
Return the minimum number of moves that you need to determine with certainty what the value of f is.

Example 1:

Input: k = 1, n = 2
Output: 2
Explanation:
Drop the egg from floor 1. If it breaks, we know that f = 0.
Otherwise, drop the egg from floor 2. If it breaks, we know that f = 1.
If it does not break, then we know f = 2.
Hence, we need at minimum 2 moves to determine with certainty what the value of f is.

Example 2:

Input: k = 2, n = 6
Output: 3

Example 3:

Input: k = 3, n = 14
Output: 4

Constraints:
- 1 <= k <= 100
- 1 <= n <= 104

Ques2: 1155. Number of Dice Rolls With Target Sum

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

Given three integers n, k, and target, return the number of possible ways (out of the kn 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 109 + 7.

Example 1:

Input: n = 1, k = 6, target = 3
Output: 1
Explanation: You throw one die with 6 faces.
There is only one way to get a sum of 3.

Example 2:

Input: n = 2, k = 6, target = 7
Output: 6
Explanation: You throw two dice, each with 6 faces.
There are 6 ways to get a sum of 7: 1+6, 2+5, 3+4, 4+3, 5+2, 6+1.

Example 3:

Input: n = 30, k = 30, target = 500
Output: 222616187
Explanation: The answer must be returned modulo 109 + 7.

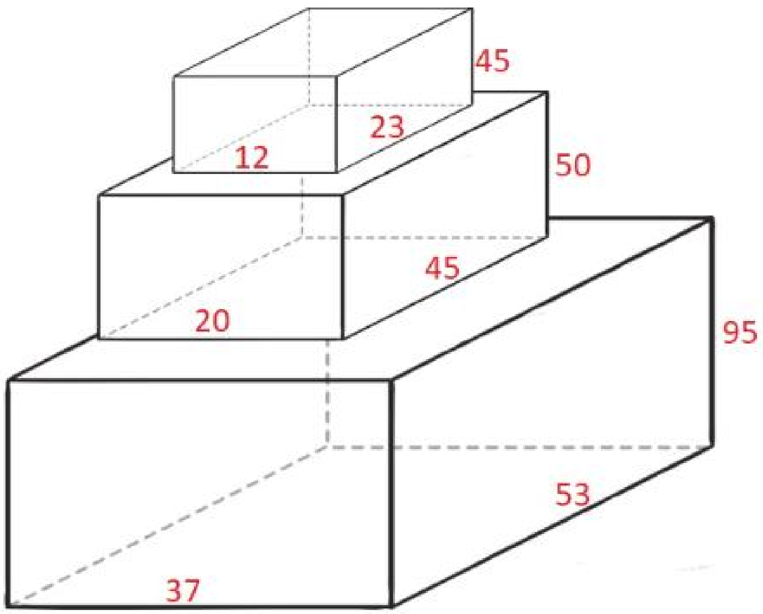
Constraints:
- 1 <= n, k <= 30
- 1 <= target <= 1000

Ques3: 1691. Maximum Height by Stacking Cuboids

Given n cuboids where the dimensions of the ith cuboid is cuboids[i] = [widthi, lengthi, heighti] (0-indexed). Choose a subset of cuboids and place them on each other.

You can place cuboid i on cuboid j if widthi <= widthj and lengthi <= lengthj and heighti <= heightj. You can rearrange any cuboid's dimensions by rotating it to put it on another cuboid.
Return the maximum height of the stacked cuboids.

Example 1:



Input: cuboids = [[50,45,20],[95,37,53],[45,23,12]]
Output: 190
Explanation:
Cuboid 1 is placed on the bottom with the 53x37 side facing down with height 95.
Cuboid 0 is placed next with the 45x20 side facing down with height 50.
Cuboid 2 is placed next with the 23x12 side facing down with height 45.
The total height is 95 + 50 + 45 = 190.

Example 2:

Input: cuboids = [[38,25,45],[76,35,3]]
Output: 76
Explanation:
You can't place any of the cuboids on the other.
We choose cuboid 1 and rotate it so that the 35x3 side is facing down and its height is 76.

Example 3:

Input: cuboids = [[7,11,17],[7,17,11],[11,7,17],[11,17,7],[17,7,11],[17,11,7]]
Output: 102
Explanation:
After rearranging the cuboids, you can see that all cuboids have the same dimension.
You can place the 11x7 side down on all cuboids so their heights are 17.
The maximum height of stacked cuboids is 6 * 17 = 102.

Constraints:
- n == cuboids.length
- 1 <= n <= 100
- 1 <= widthi, lengthi, heighti <= 100