2648. Number of Ways to Earn Points

Difficulty : Hard

https://leetcode.com/problems/number-of-ways-to-earn-points

There is a test that has `n` types of questions. You are given an integer `target` and a **0-indexed** 2D integer array `types` where `types[i] = [count_i, marks_i]` indicates that there are `count_i` questions of the `i^th` type, and each one of them is worth `marks_i` points.

Return *the number of ways you can earn* ***exactly*** `target` *points in the exam*. Since the answer may be too large, return it **modulo** `10^9 + 7`.

Note that questions of the same type are indistinguishable.

- For example, if there are `3` questions of the same type, then solving the `1^st` and `2^nd` questions is the same as solving the `1^st` and `3^rd` questions, or the `2^nd` and `3^rd` questions.

Example 1:

```
Input: target = 6, types = [[6,1],[3,2],[2,3]]
Output: 7
Explanation: You can earn 6 points in one of the seven ways:
- Solve 6 questions of the 0th type: 1 + 1 + 1 + 1 + 1 + 1 = 6
- Solve 4 questions of the 0th type and 1 question of the 1st type: 1 + 1 + 1 + 1 + 2 = 6
- Solve 2 questions of the 0th type and 2 questions of the 1st type: 1 + 1 + 2 + 2 = 6
- Solve 3 questions of the 0th type and 1 question of the 2nd type: 1 + 1 + 1 + 3 = 6
- Solve 1 question of the 0th type, 1 question of the 1st type and 1 question of the 2nd type: 1 + 2 + 3 = 6
- Solve 3 questions of the 1st type: 2 + 2 + 2 = 6
- Solve 2 questions of the 2nd type: 3 + 3 = 6
```

Example 2:

```
Input: target = 5, types = [[50,1],[50,2],[50,5]]
Output: 4
Explanation: You can earn 5 points in one of the four ways:
- Solve 5 questions of the 0th type: 1 + 1 + 1 + 1 + 1 = 5
- Solve 3 questions of the 0th type and 1 question of the 1st type: 1 + 1 + 1 + 2 = 5
- Solve 1 questions of the 0th type and 2 questions of the 1st type: 1 + 2 + 2 = 5
- Solve 1 question of the 2nd type: 5
```

Example 3:

```
Input: target = 18, types = [[6,1],[3,2],[2,3]]
Output: 1
Explanation: You can only earn 18 points by answering all questions.
```

Constraints:

- `1 <= target <= 1000`
- `n == types.length`
- `1 <= n <= 50`
- `types[i].length == 2`
- `1 <= count_i, marks_i <= 50`