

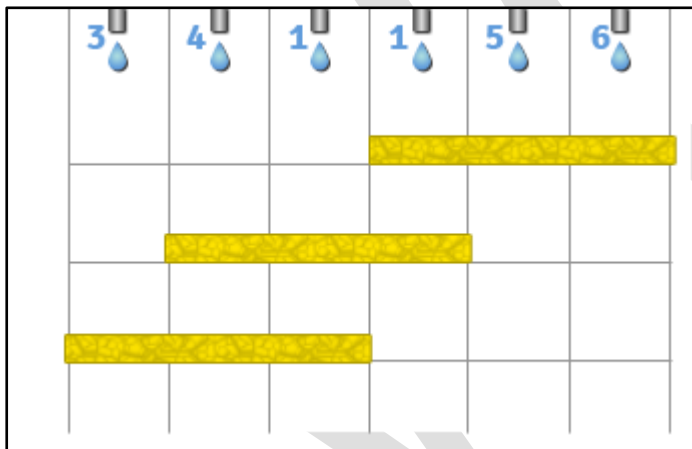
WCC : DYNAMIC PROGRAMMING

PROBLEM STATEMENT:

Manao is conducting an experiment to measure the humidity level of some spongy material. The experiment is conducted in a closed room which is observed from a side, so we will consider it in two dimensions.

The room is N centimeters wide. There are N drop counters built in the ceiling of the room. Drop counter 0 is attached 0.5 centimeters from the left end of the room, and each next one is attached 1 centimeter to the right of the previous one. You are given a `String[]` `intensity` filled with digits. Concatenate its elements to obtain one `String` `S` of length N . The i -th (0-based index) drop counter drips exactly `S[i]` drops per minute, where `S[i]` is the digit at the i -th position in `S`.

Manao is going to place M sponges in the room. All sponges are exactly L centimeters long and their thickness is negligible. Each sponge must be placed horizontally, it must be entirely within the room, and the distance between the left wall of the room and the sponge must be an integer. In other words, the coordinate of its left end must be an integer between 0 and $N-L$, inclusive. Different sponges must be attached at different heights. Each sponge will totally absorb every drop that drips at it. For example, in the following picture, the sponges (from top to bottom) absorb 12, 5, and 3 drops per minute.



The experiment requires the sponges to be attached in such a way that each of them absorbs between A and B drops per minute, inclusive. Manao is interested in the number of ways in which this can be done. Two ways P and Q of attaching the sponges are the same if and only if for each sponge S_1 in P there exists a sponge S_2 in Q such that S_1 and S_2 lie directly beneath the same sets of drop counters. A sponge X lies directly beneath a drop counter Y if there is no other sponge between them. That is, if there is some water dripping from the drop counter Y , it all lands on the sponge X . Note that according to these definitions the sponges are indistinguishable.

You are given the `String[]` `intensity` and the ints M , L , A , and B . Count the number of different ways to attach the sponges and return that count modulo 1,000,000,009.

Definition

Class: TheExperiment
Method: countPlacements
Parameters: String[], int, int, int, int
Returns: int
Method signature: int countPlacements(String[] intensity, int M, int L, int A, int B)
(be sure your method is public)

Constraints

- intensity will contain between 1 and 6 elements, inclusive.
- Each element of intensity will be between 1 and 50 characters long, inclusive.
- Each element of intensity will consist of digits ('0'-'9') only.
- M will be between 1 and 300, inclusive.
- L will be between 1 and N, inclusive, where N is the total number of characters in intensity.
- A will be between 1 and 2700, inclusive.
- B will be between A and 2700, inclusive.

Examples Test Cases:

0)

{"341156"}

3

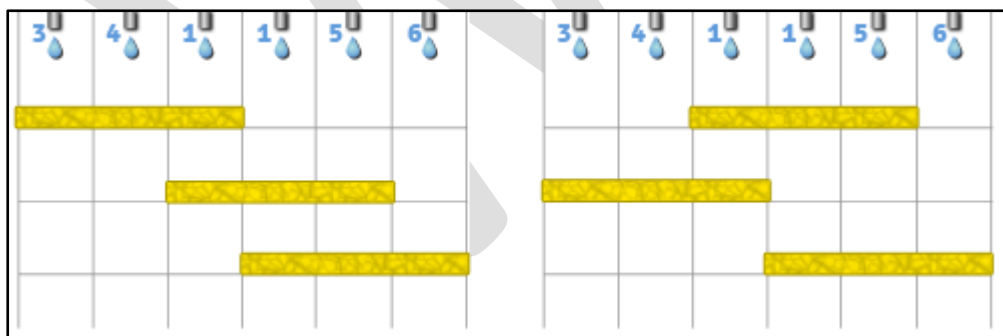
3

6

10

Returns: 2

Manao needs to place three sponges of length 3 in such a way that they absorb between 6 and 10 drops per minute. The valid ways to attach the sponges are the following:



In the first way, the sponges receive 8, 6 and 6 drops, from highest to lowest. In the second way, they absorb 7, 7 and 6. Note that Manao could interchange the heights of the lower two sponges in the second picture, but this would result in the same way.

