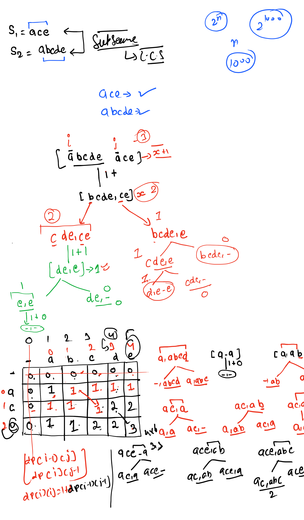


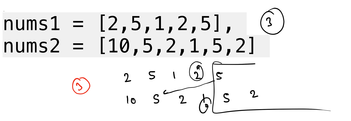
You are given two integer arrays `nums1` and `nums2`. We write the integers of `nums1` and `nums2` (in the order they are given) on two separate horizontal lines.

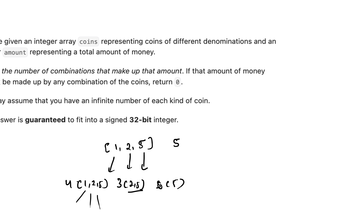
We may draw connecting lines: a straight line connecting two numbers `nums1[i]` and `nums2[j]` such that:

- `nums1[i] == nums2[j]`, and
- the line we draw does not intersect any other connecting (non-horizontal) line.

Note that a connecting line cannot intersect even at the endpoints (i.e., each number can only belong to one connecting line).

Return *the maximum number of connecting lines we can draw in this way*.



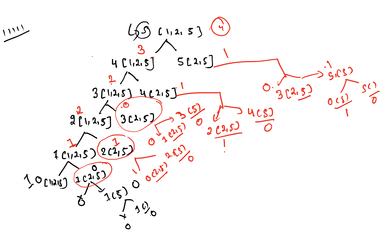


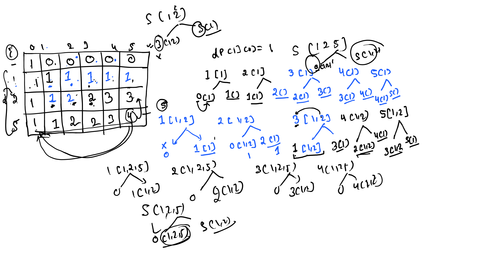
You are given an integer array `coins` representing coins of different denominations and an integer `amount` representing a total amount of money.

Return *the number of combinations that make up that amount*. If that amount of money cannot be made up by any combination of the coins, return `0`.

You may assume that you have an infinite number of each kind of coin.

The answer is **guaranteed** to fit into a signed **32-bit** integer.



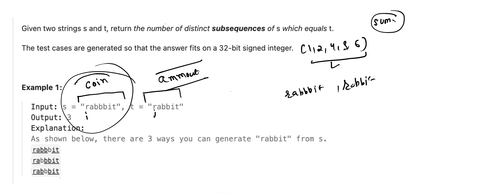


Given two strings s and t, return *the number of distinct* ***subsequences*** *of* s *which equals* t.

The test cases are generated so that the answer fits on a 32-bit signed integer.

Example 1:

```
Input: s = "rabbbit", t = "rabbit"
Output:
Explanation:
As shown below, there are 3 ways you can generate "rabbit" from s.
rabbbit
rabbbit
rabbbit
```



You are given two strings, A and B. Answer, what is the smallest number of operations you need to transform A to B?

Operations are:

1. Delete one letter from one of strings
2. Insert one letter into one of strings
3. Replace one of letters from one of strings with another letter

Input

T - number of test cases

For each test case:

- String A
- String B

Both strings will contain only uppercase characters and they won't be longer than 2000 characters.

There will be 10 test cases in data set.