2466. Count Ways To Build Good Strings

Medium | Topics | Companies | Hint

Given the integers `zero`, `one`, `low`, and `high`, we can construct a string by starting with an empty string, and then at each step perform either of the following:

- Append the character `'0'` `zero` times.
- Append the character `'1'` `one` times.

This can be performed any number of times.

A **good** string is a string constructed by the above process having a **length** between `low` and `high` (**inclusive**).

Return *the number of **different** good strings that can be constructed satisfying these properties.* Since the answer can be large, return it **modulo** $10^9 + 7$.

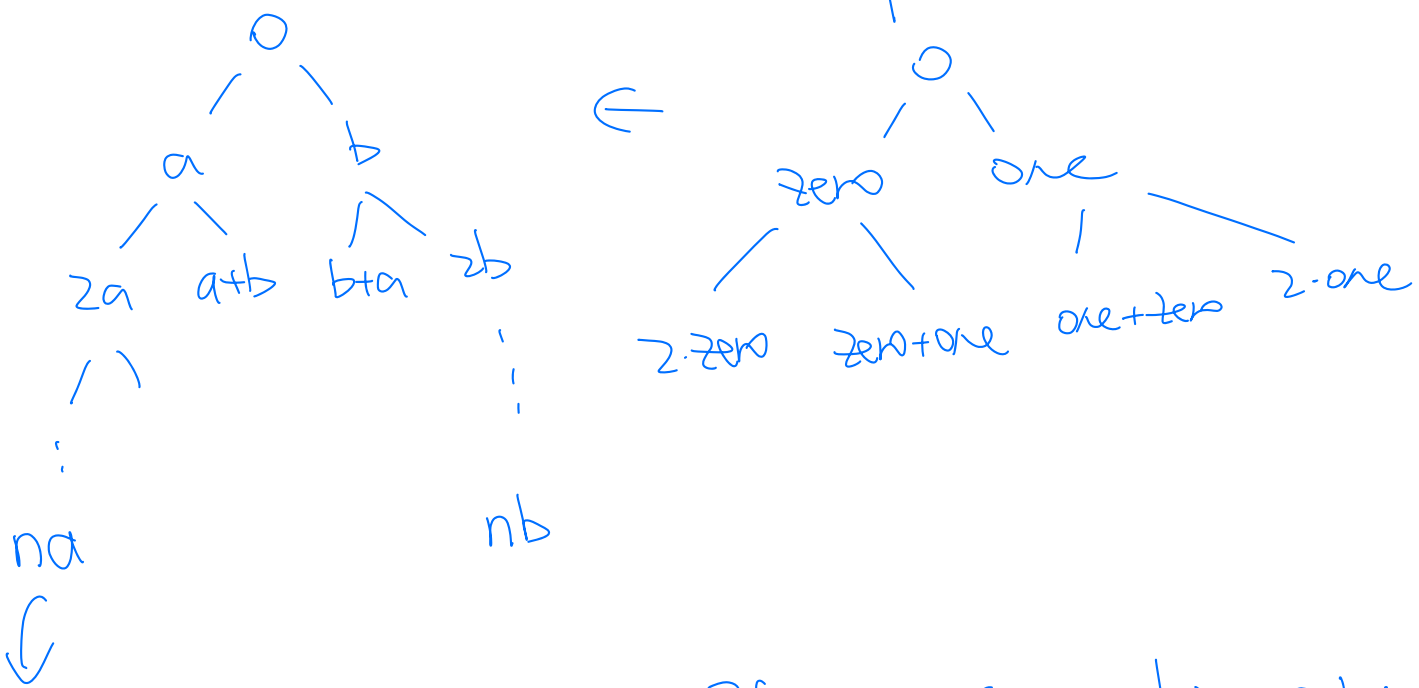
Example 1:

```
Input: low = 3, high = 3, zero = 1, one = 1
Output: 8
Explanation:
One possible valid good string is "011".
It can be constructed as follows: "" -> "0" -> "01" -> "011".
All binary strings from "000" to "111" are good strings in this example.
```

① DP

② Backtracking.

最后 length



$low \le \min(na, nb)$ & $\max(ma, mb) \le high$

$n = low$ / $\max(zero, one)$

$m = high$ / $\min(zero, one)$

$n \to m$?

③ NeetCode.io

```python
class Solution:
    def countGoodStrings(self, low: int, high: int, zero: int, one: int) -> int:

        mod = 10**9 + 7
        dp = {}

        def dfs(length):
            if length > high:
                return 0
            if length in dp:
                return dp[length]

            dp[length] = 1 if length >= low else 0
            dp[length] += dfs(length + zero) + dfs(length + one)
            return dp[length] % mod

        return dfs(0)
```

```python
class Solution:
    def countGoodStrings(self, low: int, high: int, zero: int, one: int) -> int:
        dp = { 0: 1 } # length -> num of strs
        mod = 10**9 + 7

        for i in range(1, high + 1):
            dp[i] = (dp.get(i - one, 0) + dp.get(i - zero, 0)) % mod

        return sum([dp[i] for i in range(low, high + 1)])

```