

Problem 902: Numbers At Most N Given Digit Set

Problem Information

Difficulty: Hard

Acceptance Rate: 44.10%

Paid Only: No

Tags: Array, Math, String, Binary Search, Dynamic Programming

Problem Description

Given an array of `digits` which is sorted in **non-decreasing** order. You can write numbers using each `digits[i]` as many times as we want. For example, if `digits = ['1','3','5']`, we may write numbers such as `13`, `551`, and `1351315`.

Return the number of positive integers that can be generated that are less than or equal to a given integer `n`.

Example 1:

Input: digits = ["1","3","5","7"], n = 100 **Output:** 20 **Explanation:** The 20 numbers that can be written are: 1, 3, 5, 7, 11, 13, 15, 17, 31, 33, 35, 37, 51, 53, 55, 57, 71, 73, 75, 77.

Example 2:

Input: digits = ["1","4","9"], n = 1000000000 **Output:** 29523 **Explanation:** We can write 3 one digit numbers, 9 two digit numbers, 27 three digit numbers, 81 four digit numbers, 243 five digit numbers, 729 six digit numbers, 2187 seven digit numbers, 6561 eight digit numbers, and 19683 nine digit numbers. In total, this is 29523 integers that can be written using the digits array.

Example 3:

Input: digits = ["7"], n = 8 **Output:** 1

Constraints:

* `1` <= digits.length <= 9` * `digits[i].length == 1` * `digits[i]` is a digit from `1` to `9`. * All the values in `digits` are **unique**. * `digits` is sorted in **non-decreasing** order. * `1` <= n <= 109`

Code Snippets

C++:

```
class Solution {
public:
    int atMostNGivenDigitSet(vector<string>& digits, int n) {

    }
};
```

Java:

```
class Solution {
    public int atMostNGivenDigitSet(String[] digits, int n) {

    }
}
```

Python3:

```
class Solution:
    def atMostNGivenDigitSet(self, digits: List[str], n: int) -> int:
```