Previous Next

171. Excel Sheet Column Number

June 22, 2020 | 2.1K views

Average Rating: 4.50 (8 votes)

Given a column title as appear in an Excel sheet, return its corresponding column number.

For example:

```
A -> 1
B -> 2
C -> 3
...
Z -> 26
AA -> 27
AB -> 28
...
```

Example 1:

```
Input: "A"
Output: 1
```

Example 2:

```
Input: "AB"
Output: 28
```

Example 3:

```
Input: "ZY"
Output: 701
```

Constraints:

- `1 <= s.length <= 7`
- `s` consists only of uppercase English letters.
- `s` is between "A" and "FXSHRXW".

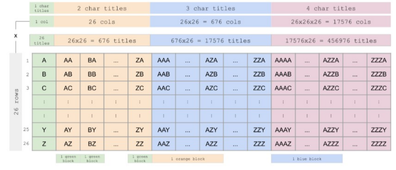
Solution

This problem can be solved as if it is a problem of converting base-26 number system to base-10 number system.

Approach 1: Right to Left

Intuition

Let's tabulate the titles of an excel sheet in a table. There will be 26 rows in each column. Each cell in the table represents an excel sheet title.



Pay attention to the "1 green block", "1 orange block" and "1 blue block" in the figure. These tell how bigger blocks are composed of smaller blocks. For example, blocks of 2-character titles are composed of 1-character blocks and blocks of 3-character titles are composed of 2-character blocks. This information is useful for finding a general pattern when calculating the values of titles.

Let's say we want to get the value of title **AZZC**. This can be broken down as `'A***' + 'Z**' + 'Z*' + 'C'`. Here, the `*`s represent smaller blocks. `*` means a block of 1-character titles. `**` means a block of 2-character titles. There are 26^1 titles in a block of 1-character titles. There are 26^2 titles in a block of 2-character titles.

Scanning **AZZC** from right to left while accumulating results:

1. First, ask the question, what the value of `'C'` is:
   - `'C'` = 3 x 26^0 = 3 x 1 = 3
   - `result` = 0 + 3 = **3**
2. Then, ask the question, what the value of `'Z*'` is:
   - `'Z*'` = 26 x 26^1 = 26 x 26 = 676
   - `result` = **3** + 676 = **679**
3. Then, ask the question, what the value of `'Z**'` is:
   - `'Z**'` = 26 x 26^2 = 26 x 676 = 17576
   - `result` = **679** + 17576 = **18255**
4. Finally, ask the question, what the value of `'A***'` is:
   - `'A***'` = 1 x 26^3 = 1 x 17576 = 17576
   - `result` = **18255** + 17576 = 35831

Algorithm

- To get indices of alphabets, create a mapping of alphabets and their corresponding values. (1-indexed)
- Initialize an accumulator variable `result`.
- Starting from right to left, calculate the value of the character associated with its position and add it to `result`.

Implementation

Java Python3 Copy

```python
class Solution:
    def titleToNumber(self, s: str) -> int:
        result = 0

        # Decimal 65 in ASCII corresponds to char 'A'
        alpha_map = {chr(i + 65): i + 1 for i in range(26)}

        n = len(s)
        for i in range(n):
            cur_char = s[n - 1 - i]
            result += (alpha_map[cur_char] * (26 ** i))
        return result
```

Complexity Analysis

- Time complexity : $O(N)$ where N is the number of characters in the input string.
- Space complexity : $O(1)$. Even though we have an alphabet to index mapping, it is always constant.

Approach 2: Left to Right

Intuition

Rather than scanning from right to left as described in Approach 1, we can also scan the title from left to right.

For example, if we want to get the decimal value of string "1337", we can iteratively find the result by scanning the string from left to right as follows:

1. '1' = **1**
2. '13' = (**1** x 10) + 3 = **13**
3. '133' = (**13** x 10) + 3 = **133**
4. '1337' = (**133** x 10) + 7 = 1337

Instead of base-10, we are dealing with base-26 number system. Based on the same idea, we can just replace 10s with 26s and convert alphabets to numbers.

For a title "LEET":

1. L = **12**
2. E = (**12** x 26) + 5 = **317**
3. E = (**317** x 26) + 5 = **8247**
4. T = (**8247** x 26) + 20 = 214442

In Approach 1, we have built a mapping of alphabets to numbers. There is another way to get the number value of a character without building an alphabet mapping. You can do this by converting a character to its ASCII value and subtracting ASCII value of character 'A' from that value. By doing so, you will get results from 0 (for A) to 25 (for Z). Since we are indexing from 1, we can just add 1 up to the result. This eliminates a loop where you create an alphabet to number mapping which was done in Approach 1.

Implementation

Java Python3 Copy

```python
class Solution:
    def titleToNumber(self, s: str) -> int:
        result = 0
        n = len(s)
        for i in range(n):
            result = result * 26
            result += (ord(s[i]) - ord('A') + 1)
        return result
```

Complexity Analysis

- Time complexity : $O(N)$ where N is the number of characters in the input string.
- Space complexity : $O(1)$.

Rate this article:

Previous Next

Comments: 2

Sort By

Type comment here... (Markdown is supported)

Preview Post

maggiez98 ★ 1 June 23, 2020 11:34 PM

The question/solution didn't address any edge cases - when the input contains non-letters and when the int overflows. I also googled excel sheet and apparently they only have a limited number of columns. So the question should have made these clearer.

1 Share Reply

SHOW 1 REPLY

hegdeShai ★ 4 July 4, 2020 7:44 PM

This is definitely not an easy problem!

0 Share Reply

SHOW 2 REPLIES