
Daily Coding Problem #212

Problem

This problem was asked by Dropbox.

Spreadsheets often use this alphabetical encoding for its columns: "A", "B", "C", ..., "AA", "AB", ..., "ZZ", "AAA", "AAB",

Given a column number, return its alphabetical column id. For example, given 1, return "A". Given 27, return "AA".

Solution

Let's look at some simpler cases first and see if we can find a pattern.

First, what would we do if the input was between 1 and 26? In this case, we can map our input using the dictionary {1: "A", 2: "B", ..., 26: "Z"}.

Now let's look at some two-letter solutions. What would happen if our input was between 27 and 52? In this case, our first letter would be "A", and our second letter would be mapped as before. What if our input was between 53 and 78? This would be similar, except our first letter would be "B".

More mathematically, for any two-letter outcome, we can perform the following steps:

- Floor divide $n - 1$ by 26
- To get the first letter, map the quotient using our dictionary
- To get the second letter, map the remainder using our dictionary

What if the result of our floor division is greater than 26? In this case, it is impossible to return a two-letter solution. Generalizing the pattern above, we can continue dividing our number by 26 and prepending the remainder to our solution, until we reach 0.

To give an example, here is how we would deal with the input 1000:

- Divide 999 / 26 to get 38, with a remainder of 11. Solution: "L" + "".
- Divide 37 / 26 to get 1, with a remainder of 11. Solution: "L" + "L".
- Divide 1 / 26 to get 0, with a remainder of 1. Solution: "A" + "LL".

As a result, we would return "ALL".

Here's how this could look in code:

```
def encode(n):  
    s = ""  
    while n > 0:  
        n, remainder = divmod(n - 1, 26)  
        s = chr(65 + remainder) + s  
    return s
```

This solution is $O(N)$, where N is the number of digits in our input.

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)