

273. 整数转换英文表示 (AMZ)

方法一：递归

由于非负整数 num 的最大值为 $2^{31}-1$ 因此最多有 10 位数。将整数转换成英文表示中，将数字按照 3 位一组划分，将每一组的英文表示拼接之后即可得到整数 num 的英文表示。

每一组最多有 3 位数，可以使用递归的方式得到每一组的英文表示。根据数字所在的范围，具体做法如下：

小于 20 的数可以直接得到其英文表示；

大于等于 20 且小于 100 的数首先将十位转换成英文表示，然后对个位递归地转换成英文表示；

大于等于 100 的数首先将百位转换成英文表示，然后对其余部分（十位和个位）递归地转换成英文表示。

从高到低的每一组的单位依次是 10^9 、 10^6 、 10^3 、1，除了最低组以外，每一组都有对应的表示单位的词，分别是“Billion”、“Million”、“Thousand”。

得到每一组的英文表示后，需要对每一组加上对应的表示单位的词，然后拼接得到整数 num 的英文表示。

具体实现中需要注意以下两点：

只有非零的组的英文表示才会拼接到整数 num 的英文表示中；

如果 $num=0$ ，则不适用上述做法，而是直接返回“Zero”。//

<https://leetcode-cn.com/problems/integer-to-english-words/solution/zheng-shu-zhuan-huan-ying-wen-biao-shi-b-ivik/>

笔者在去年夏天的微软面试中碰到过原题。面试官要求用递归和迭代都实现一遍。这里讲解我更喜欢递归写法。

英语世界里数字喜欢用逗号分隔，每三位一组，有一个对应的单位。从低到高分别是千 (Thousand), 百万 (million), 十亿 (billion)；每个单位前最多有一个小于 1000 的数做量词，我们要做的只是找到一个计算量词的方式（1000 以下的数字如何转化成英文），并拼接这些量词和单位即可。

比如 1234567891 可以写成 1,234,567,891

代表

1 个 亿

234 个 百万

567 个 千

891

1000 以下的数字如何表示呢？

比如517

代表

5个100 + 17

我们知道英语20以下都有直接对应的单词。20以上写成几十+几。

<https://leetcode-cn.com/problems/integer-to-english-words/solution/wei-rao-li-lun-mo-ni-ti-ying-wen-shu-zi-1sc7b/>

<https://www.youtube.com/watch?v=R6BiEJG4BZ0>

```
var numberToWords = function(num) {  
    if (num === 0) return 'Zero'  
  
    let billion = Math.floor(num / 1e9)  
    let million = Math.floor(num / 1e6) % 1e3  
    let thousand = Math.floor(num / 1e3) % 1e3  
    let hundred = Math.floor(num / 100) % 10  
    let ten = Math.floor(num / 10) % 10  
    let one = num % 10  
  
    let Tens = ['', '', 'Twenty', 'Thirty', 'Forty', 'Fifty', 'Sixty',  
    'Seventy', 'Eighty', 'Ninety']  
    let Teens = ['Ten', 'Eleven', 'Twelve', 'Thirteen', 'Fourteen',  
    'Fifteen', 'Sixteen', 'Seventeen', 'Eighteen', 'Nineteen']  
    let ones = ['Zero', 'One', 'Two', 'Three', 'Four', 'Five', 'Six',  
    'Seven', 'Eight', 'Nine']  
  
    let res = ''  
    if (billion) res += numberToWords(billion) + ' Billion'  
    if (million) res += ' ' + numberToWords(million) + ' Million'  
    if (thousand) res += ' ' + numberToWords(thousand) + ' Thousand'  
    if (hundred) res += ' ' + ones[hundred] + ' Hundred'  
    if (ten > 1) {  
        res += ' ' + Tens[ten]  
        if (one) res += ' ' + ones[one]  
    } else if (ten === 1) {  
        res += ' ' + Teens[one]  
    } else if (one > 0) {  
        res += ' ' + ones[one]  
    }  
    return res.trim()  
}
```

```
};
```

```
var numberToWords = function(num) {  
    const singles = ["", "One", "Two", "Three", "Four", "Five", "Six",  
"Seven", "Eight", "Nine"];  
    const teens = ["Ten", "Eleven", "Twelve", "Thirteen", "Fourteen",  
"Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"];  
    const tens = ["", "Ten", "Twenty", "Thirty", "Forty", "Fifty", "Sixty",  
"Seventy", "Eighty", "Ninety"];  
    const thousands = ["", "Thousand", "Million", "Billion"];  
  
    const recursion = (curr, num) => {  
        if (num === 0) {  
            return;  
        } else if (num < 10) {  
            curr.push(singles[num] + " ");  
        } else if (num < 20) {  
            curr.push(teens[num - 10] + " ");  
        } else if (num < 100) {  
            curr.push(tens[Math.floor(num / 10)] + " ");  
            recursion(curr, num % 10);  
        } else {  
            curr.push(singles[Math.floor(num / 100)] + " Hundred ");  
            recursion(curr, num % 100);  
        }  
    }  
  
    if (num === 0) {  
        return "Zero";  
    }  
    const sb = [];  
    for (let i = 3, unit = 1000000000; i >= 0; i--, unit = Math.floor(unit  
/ 1000)) {  
        const curNum = Math.floor(num / unit);  
        if (curNum !== 0) {  
            num -= curNum * unit;  
            const curr = [];  
            recursion(curr, curNum);  
            curr.push(thousands[i] + " ");  
            sb.push(curr.join(''));  
        }  
    }  
    return sb.join('').trim();  
}
```

```
}
```

```
//
```

```
https://leetcode-cn.com/problems/integer-to-english-words/solution/zheng-shu-zhuan-huan-ying-wen-biao-shi-b-ivik/
```