

## 一种字符串和数字的对应关系

### 【题目】

一个 `char` 类型的数组 `chs`，其中所有的字符都不同。

例如，`chs=['A','B','C',... 'Z']`，则字符串与整数的对应关系如下：

A, B... Z, AA, AB... AZ, BA, BB... ZZ, AAA... ZZZ, AAAA...

1, 2... 26, 27, 28... 52, 53, 54... 702, 703... 18278, 18279...

例如，`chs=['A','B','C']`，则字符串与整数的对应关系如下：

A, B, C, AA, AB... CC, AAA... CCC, AAAA...

1, 2, 3, 4, 5... 12, 13... 39, 40...

给定一个数组 `chs`，实现根据对应关系完成字符串与整数相互转换的两个函数。

### 【难度】

校 ★★★☆

### 【解答】

面试者在分析本题时，往往会将字符串与数字的对应关系与  $K$  进制数联系起来， $K$  指 `chs` 的长度，比如，第一个例子中 `chs` 的长度为 26。最终会发现用  $K$  进制数是不能实现的。下面就解释一下本题的对应关系与  $K$  进制数不同的地方。

$K$  进制数是每一个位置上的值只能在  $[0, K-1]$  之间取值。例如，十进制数的 72，高位为 7，低位为 2。十进制数的 72 转换成三进制数的表达为 “2200”，也就是  $72=27\times 2+9\times 2+3\times 0+1\times 0$ 。但是本题描述的对应方式却不是这样，我们暂时把题目描述的对应方式叫作  $K$  伪进制数， $K$  伪进制数是每一个位置上的值只能在  $[1, K]$  之间取值。以 `chs=['A','B','C']` 来举例，即 3 伪进制数。如果把十进制数的 72 用这个 `chs` 的 3 伪进制数表示，是 “BABC”，也就是  $72=27\times 2+9\times 1+3\times 2+1\times 3$ 。也就是对  $K$  进制数来讲，每个位（如：27、9、3、1）上的值是可以取 0 的，但如果位上的值不为 0，也在  $[1, K-1]$  范围上。而对  $K$  伪进制数来讲，每个位上的值绝对不能取 0，而是必须在  $[1, K]$  之间。所以用  $K$  进制的思路是不能实现本题的对应关系的。

下面解释一下本书提供的解法，先看从数字如何得到字符串。还是以 `chs=['A','B','C']`

来举例，以下是十进制数的 72 得到表达它的字符串的过程：

1. chs 的长度为 3，所以这是一个 3 伪进制，从低位到高位依次为 1, 3, 9, 27, 81...

2. 从 1 开始减，72 减去 1，剩下 71；71 减去 3，剩下 68；68 减去 9，剩下 59；59 减去 27，剩下 32；32 减去 81 时，发现不够减，此时就知道想要表达十进制数的 72，只需使用 3 伪进制的前 4 位，也就是 27, 9, 3, 1，而不必扩到第 5 位的 81。换句话说，既然  $K$  伪进制中每个位上的值都不能为 0，就从低位到高位把每个位置上的值都先减去 1 遍，看这个数到底需要前几位。

3. 步骤 2 剩下的数是 32，同时前四位的值已经使用了 1 次，即  $72 - 32 = 40 = 27 \times 1 + 9 \times 1 + 3 \times 1 + 1 \times 1 = \text{"AAAA"}$ 。接下来看剩下的 32 最多可以用几个 27 呢？最多用 1 个 ( $32/27=1$ )，再算上之前的一个 27，一共要 2 个 27 (B)。  $32\%27$  的结果是 5，这表示让 32 减去尽量多的 27 而剩下来的数。然后看 5 最多可以用几个 9，一个也用不了，再算上之前的一个 9，一共要 1 个 9 (A)。  $5\%9=5$ ，接下来看 5 最多可以用几个 3，1 个，再算上之前的一个 3，一共要 2 个 3 (B)。  $5\%3=2$ ，最后看 2 最多可以用几个 1，2 个，算上之前的一个 1，一共 3 个 1 (C)。所以结果是 "BABC"。

上文所描述的  $K$  伪进制虽然和  $K$  进制不同，但是把十进制数转换成  $K$  伪进制数的过程却和把十进制数转换成  $K$  进制数的过程相似。具体说来，步骤 2 中是从低位到高位看一个数  $N$  最多用几个  $K$  伪进制的位，时间复杂度为  $O(\log N)$  (以  $K$  为底)，步骤 3 是从高位到低位反着回去看每个位上的值最多是多少，时间复杂度也是  $O(\log N)$  (以  $K$  为底)， $K$  为 chs 的长度，所以以上过程的时间复杂度为  $O(\log N)$  (以 chs 的长度为底)。

数字到字符串的全部过程请参看如下代码中的 getString 方法。

```
public String getString(char[] chs, int n) {
    if (chs == null || chs.length == 0 || n < 1) {
        return "";
    }
    int cur = 1;
    int base = chs.length;
    int len = 0;
    while (n >= cur) {
        len++;
        n -= cur;
        cur *= base;
    }
    char[] res = new char[len];
    int index = 0;
    int nCur = 0;
    do {
        cur /= base;
        nCur = n / cur;
```

```

        res[index++] = getKthCharAtChs(chs, nCur + 1);
        n %= cur;
    } while (index != res.length);
    return String.valueOf(res);
}

public char getKthCharAtChs(char[] chs, int k) {
    if (k < 1 || k > chs.length) {
        return 0;
    }
    return chs[k - 1];
}

```

接下来介绍如何通过字符串得到对应的数字。其实如果理解了  $K$  伪进制数的含义，算出字符串对应的数字就十分容易了。例如，`chs=['A','B','C']`，字符串是"ABBA"，可以知道这个字符串的含义是 27 有 1 个，9 有 2 个，3 有 2 个，1 有 1 个，所以对应的数字是 52。具体过程请参看如下代码中的 `getNum` 方法。

```

public int getNum(char[] chs, String str) {
    if (chs == null || chs.length == 0) {
        return 0;
    }
    char[] strc = str.toCharArray();
    int base = chs.length;
    int cur = 1;
    int res = 0;
    for (int i = strc.length - 1; i != -1; i--) {
        res += getNthFromChar(chs, strc[i]) * cur;
        cur *= base;
    }
    return res;
}

public int getNthFromChar(char[] chs, char ch) {
    int res = -1;
    for (int i = 0; i != chs.length; i++) {
        if (chs[i] == ch) {
            res = i + 1;
            break;
        }
    }
    return res;
}

```