

第一周习题刷题记录

961.重复N次的元素

题目：在大小为 $2N$ 的数组 A 中有 $N+1$ 个不同的元素，其中有一个元素重复了 N 次。

返回重复了 N 次的那个元素。

算法：

1. 哈希表，此时会有额外的空间消耗
2. 线性遍历
 - 由于题目中说了必定有 $N+1$ 个不同数字，我们考虑将整个数组划分为
$$(a_1, a_2), (a_3, a_4), \dots, (a_{n-1}, a_n)(a_1, a_2), (a_3, a_4), \dots, (a_{n-1}, a_n)$$
 - 如果任意一组相同，则改数就是答案，否则每组中两个数必有一个为答案

```
class Solution {
public:
    int repeatedNTimes(vector<int>& A) {
        for(int i = 0; i < A.size(); i += 2){
            if(A[i] == A[i+1]){
                return A[i];
            }
        }
        if(A[0] == A[2] || A[0] == A[3]){
            return A[0];
        }
        return A[1];
    }
};
```

394.字符串解码

题目：给定一个经过编码的字符串，返回它解码后的字符串。

编码规则为: $k[\text{encoded_string}]$ ，表示其中方括号内部的 encoded_string 正好重复 k 次。注意 k 保证为正整数。

你可以认为输入字符串总是有效的；输入字符串中没有额外的空格，且输入的方括号总是符合格式要求的。

此外，你可以认为原始数据不包含数字，所有的数字只表示重复的次数 k ，例如不会出现像 $3a$ 或 $2[4]$ 的输入。

算法：

1. 递归
 - 如果遇到的是字母，则直接添加到答案中
 - 如果是遇到 $k[\text{encoded_string}]$ ，则解析出 k 与 encoded_string ，进行递归

```
class Solution {
public:
    string decodeString(string s) {
```

```

string res;
for (int i = 0; i < s.size();)
{
    if (!isdigit(s[i])) res += s[i ++ ];
    else {
        int j = i;
        while (isdigit(s[j])) j ++ ;
        int t = atoi(s.substr(i, j - i).c_str());
        int k = j + 1, sum = 0;
        while (sum >= 0) {
            if (s[k] == '[') sum ++ ;
            if (s[k] == ']') sum -- ;
            k ++ ;
        }
        string str = decodeString(s.substr(j + 1, k - j - 2));
        while (t -- ) res += str;
        i = k;
    }
}
return res;
};

```