

# Problem I Rank LED题解

第一届广西大学生编程大赛



## 问题 I 对 LED 进行排序

【问题描述】 主持人已布排名。每个人的排名都显示在LED屏幕上,就像我们的图片一样。



可以看到 '0' 到 '9' 的数字亮线依次为{6,2,5,5,4,5,6,3,7,6}。Luras想修改每条光线的位置,使她的新等级尽可能小,同时新等级也是一个不带任何前导零的正整数。另外,光线总数应与开始时相同。你能告诉卢拉斯她可以修改到什么最好的结果吗?

### 【输入】

第一行是一个整数T,表示案件编号。

对于每种情况,都会有 2 行。

第一行有一个整数n,它是数字串的长度。

第二行是一个数字字符串,它是一个长度为 n 的非前导零大正整数。

保证 T约为100。

对于 100% 的情况,  $1 \leq n \leq 100$ 。字符串都是非前导零数字字符串。

### 【输出】

对于每种情况,都需要输出一行。行中应该有一个非前导零正数字字符串,这意味着 Luras 可以修改到的最佳排名。

### 【样本输入】

```
3
1
9
2
99
5
10000
```

### 【输出样例】

```
6
28
2888
```

## 题目大意

'0'到'9'的数字亮线依次为{6、2、5、5、4、5、6、3、7、6}。

Luras想修改每条光线的位置,使她的新等级尽可能小,同时新等级也是一个不带任何前导零的正整数。

另外，光线总数应与开始时相同。

## 官方题解

### Problem I

题目：LED排名

题目类型：贪心 or DP

题意：给定一个LED的数字，让你修改成笔画数相同的最小正整数。

题解：首先，可以DP，用 $dp[len][sum]$ 表示数字长度恰好为 $len$ ，数字笔画数恰好为 $sum$ 是否可行。这个dp数组可以在 $O(len^2)$ 左右的复杂度内预处理。然后对于给定的初始串，其长度为 $n$ ，笔画和为 $sum$ ，找到一个最小的 $m \leq n$ ，使得 $dp[m][sum] == true$ ，然后以此为基础，从高位到低位使用尽可能小的数字，不断dfs即可。

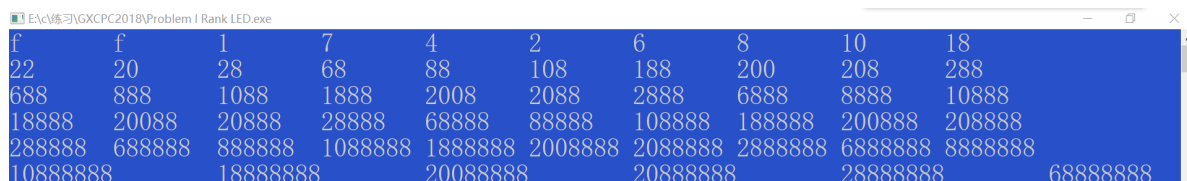
## 思路

感谢 2023牛客寒假集训营训练营3 QQ群的 [文件]第十五届蓝桥杯国赛获奖名单.pdf 同学的思路和DP代码

这题一开始想的是无脑换成8结尾，但发现不行，比如输入 23 输出 22

还是得用dp来解，当然可以尝试打表看看有没有规律啥的

使用 **dp** 通过打表发现规律（发现208及其之后的都是以8结尾）



f	f	1	7	4	2	6	8	10	18
22	20	28	68	88	108	188	200	208	288
688	888	1088	1888	2008	2088	2888	6888	8888	10888
18888	20088	20888	28888	68888	88888	108888	188888	200888	208888
288888	688888	888888	1088888	1888888	2008888	2088888	2888888	6888888	8888888
10888888	18888888	20088888	20888888	28888888	68888888	88888888	108888888	188888888	200888888

所以可以用比较偷懒的写法：把208（灯管长度18）之前的保存起来，当特例判断。之后的就直接贪（只要判断其中几个数字就行），直接从后往前输出

## dp代码（官方题解思路写法）

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

const int N = 7e2 + 5;
string dp[N];
int t, n;
string s;

vector<string> a = {"1", "7", "4", "2", "0", "6", "8"};
```

```

vector< int > b = { 2, 3, 4, 5, 6, 6, 7 };
int arr[10] = {6, 2, 5, 5, 4, 5, 6, 3, 7, 6};

string MinStr(string a, string b) {
    if (a[0] == 'f') {
        return b;
    }
    if (a.size() < b.size()) {
        return a;
    }
    if (a.size() > b.size()) {
        return b;
    }
    for (int i = 0; i < int(a.size()); i++) {
        if (a[i] < b[i]) {
            return a;
        }
        if (a[i] > b[i]) {
            return b;
        }
    }
    return a;
}

string Merge(string a, string x) {
    string str = "";
    if (x[0] == '0') {
        str += a[0];
        str += x;
        for (int i = 1; i < int(a.size()); i++) {
            str += a[i];
        }
        return str;
    }
    bool flag = true;
    for (int i = 0; i < int(a.size());) {
        if (flag && x[0] <= a[i]) {
            flag = false;
            str += x;
        } else {
            str += a[i++];
        }
    }
    if (flag) {
        str += x;
    }
    return str;
}

int main() {
    for (int i = 0; i <= 700; i++) {
        dp[i] = "f";
    }
    dp[2] = "1", dp[3] = "7", dp[4] = "4", dp[5] = "2", dp[6] = "6", dp[7] =
    "8";
}

```

```

        for (int i = 8; i <= 700; i++) {
            for (int j = 0; j < int(a.size()); j++) {
                dp[i] = MinStr(dp[i], Merge(dp[i - b[j]], a[j]));
            }
        }

// dp输出所有结果
// for (int i = 0; i <= 70; i++) {
//     cout << dp[i] << '\t';
//     if (i % 10 == 9) {
//         cout << endl;
//     }
// }

    cin >> t;
    while (t--) {
        cin >> n >> s;
        int x = 0;
        for (int i = 0; i < int(s.size()); i++) {
            x += arr[s[i] - '0'];
        }
        cout << dp[x] << endl;
    }
    return 0;
}

```