

1321 删数问题

题目描述

输入一个高精度的正整数 n ，去掉其中任意 s 个数字后剩下的数字按原左右次序组成一个新的正整数。编程对给定的 n 和 s ，寻找一种方案使得剩下的数字组成的新数最小。

输出新的正整数。（ n 不超过240位）

输入数据均不需判错。

输入

n

s

输出

最后剩下的最小数。

输入样例

175438

4

输出样例

13

解析

找规律1

如果要从1789543中删除一个数，使这个数最小，应该删除？答案是9。

如果要从178543中删除一个数，使整个数最小，应该删除？答案是8。

以此类推，我们应该依次删除7、5、4、3

找规律2

如果要从12343564中删除一个数，使这个数最小，应该删除？答案是第一个4

如果要从1233564中删除一个数，使整个数最小，应该删除？答案是6。

综上：

我们选取的贪心策略为：每一步总是选择一个使剩下的数最小的数字删去。

即按高位到低位的顺序搜索，若各位数字递增，则删除最后一个数字，否则，则删除第一个递减区间的首字符。很容易证明其正确性，如上图数字所示。

编码

```
#include<bits/stdc++.>

using namespace std;
string n; //定义字符串n
int s;

int main() {
    cin >> n >> s;
    int len = n.size(); //也可以用n.length()来取字符串n的长度
    while (s--> 0) {
        //枚举全部的字符
        for (int i = 0; i < len; i++) {
            //删除遇到的第一个递减序列的第一个数字
            //(若整个字符串为非递减序列, 则删去末尾的数字)
            if (n[i] > n[i + 1] || i == len - 1)
            {
                n.erase(i, 1); //把当前字符从字符串中删除
                break; //不可省略, 否则字符串会多删字符
            }
        }
    }

    //去掉前缀0, 并至少保留1个数字
    //输入的字符有可能存在前缀0
    while (n[0] == '0' && n[1]) {
        n.erase(0, 1); //删去当前字符串开头的'0'
    }

    cout << n; //输出字符串
    return 0;
}
```

逻辑航线培优教育, 信息学奥赛培训专家。

扫码添加作者获取更多内容。



