

设计测例 (Zuma)

任务描述

自从布置了 Zuma 这道题（题面见页面下方）之后，助教每天都会收到有缺陷的代码，比如在边界情况上考虑不足，或者最坏情况下时间不能满足题目要求。如果随机出测例，许多有缺陷的代码也能通过黑盒测试，这让助教非常烦心。

现在助教已经收到了 10 份代码（[点此下载](#)）

([attachment/0783/078364e5f28210724e1a332b9a952b8aa4ef8325.zip](#))，请你替助教出几份测例来鉴别每份代码有没有问题。

每一份代码都存在一定的错误：与其他 PA 题目类似，在 OJ 上评测时会出现 Accepted、Wrong Answer、Time Limit Exceeded、Memory Limit Exceeded、Runtime Error 等结果。

你的任务是，针对**每一份**代码，

1. 指出代码在评测时会出现的最严重错误类型。代码可能的出错类型及其严重程度定义如下：Wrong Answer > Runtime Error > Time Limit Exceeded = Memory Limit Exceeded。
2. 指出代码出现这个错误结果的原因。
3. 针对上述结果和原因，给出对应的测例，使得在该测例下，这份代码会因上面的原因产生这一错误结果。
4. 简要说明测例构造思路，重点在测例的构造是如何触发这一错误的。

对每份代码，解题报告需按照分点阐述以上第 1、2、4 点，对应的测例应标记为最终版本。

助教出题的时候当然是有一份保证正确但不保证复杂度的标程的啦，标程在这里[下载](#) ([attachment/6048/60480a92937b1f6280fdf32b62f2931c31d039d2.cpp](#))。

示例

推荐在阅读 Zuma 题目后再来看示例。

代码：

```
#include <iostream>
#include <string>

using namespace std;

string a;

void play(int rank, char color) {
    int left = rank;
    int right = rank;
    a.insert(a.cbegin() + rank, color);

    while (left >= 0 && a[left] == color) --left;
    left += 1;
    while (right < a.size() && a[right] == color) ++right;

    int size = right - left;
    if (size >= 3) a.erase(left, size);
}

int main() {
    getline(cin, a);
    int m = 0;
    cin >> m;

    int rank; char color;
    for (int i = 0; i < m; ++i) {
        cin >> rank >> color;
        play(rank, color);
    }

    cout << a << endl;

    return 0;
}
```

错误类型: Wrong Answer

错误原因: 没有考虑连消。

相应测例:

```
AABBAB
1
2 B
```

标准答案:

```
B
```

思路: 构造连消情况即可。测例中第一个操作在消除 B 后, 会引起 A 的连消; 而代码执行 play 消除相连的 B 后, 没有对相连的 A 作进一步消除。

测试说明

在 OJ 上测试时，你需要提交 .zip 等格式的压缩包，包含 01.in ~ 10.in，对应 10 份程序。如果提交单个 .in 文件，则该文件会作为全部 10 个程序的输入。测例中的 Windows 风格换行符 "\r\n" 都会转成 Unix 风格换行符 "\n"。

对于每个测例，如果相应的程序正常结束，则会将程序的输出与标程进行比较，来判断是否为 Wrong Answer。

你的测例需要符合题目的输入格式和数据范围。我们提供一个简易的 数据检查器（点此下载）(attachment/6048/60480a92937b1f6280fdf32b62f2931c31d039d2.cpp)，OJ 上的数据检查器与这不同；如果测例没有通过数据检查器，则评测结果为 Input Error。数据检查器不一定考虑到了所有情况，其检查结果仅供参考，不符合题面要求的测例即便侥幸通过数据检查器也不应得分。

由于硬盘空间有限，**提交的压缩包大小不能超过 20 MB，且 OJ 只保留最近提交的文件**。因此，你只能将最后一次提交标记为最终版本，取消标记最终版本后才能进行新的提交。

由于各种因素，对于程序运行时间和内存使用的统计，不同平台可能得到不同结果，会发生有些平台出现 Time Limit Exceeded 错误、Runtime Error 错误而有些平台不出现。我们保证提供的代码中存在的错误具有一定的平台无关性，例如 Time Limit Exceeded 的代码在时间复杂度上与优化版本具有明显差异。

附：Zuma

祖玛是一款曾经风靡全球的游戏，其玩法是：在一条轨道上初始排列着 n 个彩色珠子，其中任意三个相邻的珠子不会完全同色。此后，你可以发射珠子到轨道上并加入原有序列中。一旦有三个或更多同色的珠子变成相邻，它们就会立即消失。这类消除现象可能会连锁式发生，其间你将暂时不能发射珠子。



开发商最近准备为玩家写一个游戏过程的回放工具。他们已经在游戏内完成了过程记录的功能，而结算功能的实现则委托你来完成。

游戏过程的记录中，首先是轨道上初始的珠子序列，然后是玩家接下来所做的一系列操作。你的任务是，计算经过一系列操作之后的珠子序列。

输入

第 1 行是一个由大写字母 'A' ~ 'Z' 组成的字符串，表示轨道上初始的珠子序列，不同的字母表示不同的颜色。

第 2 行是一个数字 m ，表示整个回放过程共有 m 次操作。

接下来的 m 行依次对应于各次操作。每次操作由一个数字 k 和一个大写字母 Σ 描述，以空格分隔。其中， Σ 为新珠子的颜色。若插入前共有 b 颗珠子，则 $k \in [0, b]$ 表示新珠子嵌入之后（尚未发生消除之前）在轨道上的位序。

输出

输出共 1 行，给出所有操作（及可能随即发生的消除现象）之后轨道上的珠子序列。
如果最终轨道上没有珠子，则仅输出一个空行（即仅输出一个换行符）。

输入样例

```
ACCBA
5
1 B
0 A
2 B
4 C
0 A
```

输出样例

```
A
```

样例解释

各次操作后，轨道上的珠子序列分别是

```
ABCCBA
AABCCBA
AABBCCBA

A
```

数据范围

- $0 \leq n \leq 500,000$
- $0 \leq m \leq 500,000$
- 初始序列长度 $\leq 500,000$
- 输入保证珠子的插入位置合法

资源限制

- 时间限制：1.5 sec
- 空间限制：256 MB

提示

本题字母、数字混合输入，且初始序列可能为空，请使用足够鲁棒的读入方式。
[分块数组]

[列表可以 $O(1)$ 完成寻址，但是需要 $O(N)$ 完成删除和插入，链表可以 $O(1)$ 完成删除和插入，但是需要 $O(N)$ 完成寻址。我们将他们结合起来选择 \sqrt{n} 为一个中间值。]

[将 N 长度的列表以 $O(\sqrt{n})$ 长度断成 $O(\sqrt{n})$ 块，每块用链表连起来，这样链表的长度为 $O(\sqrt{n})$ 同时列表的长度为 $O(\sqrt{n})$ ，我们可以在 $O(\sqrt{n})$ 的时间范围内完成寻址插入和删除。]

[另一个需要注意的是在移动数组元素时，使用 memmove 能够大大提升程序的性能。]

UI powered by Twitter Bootstrap (<http://getbootstrap.com/>).

Tsinghua Online Judge is designed and coded by Li Ruizhe.

For all suggestions and bug reports, contact [oj\[at\]liruiizhe\[dot\]org](mailto:oj[at]liruiizhe[dot]org).