WA

Cothrax

August 20, 2019

Contents

1 Four LINES OF DEFENSE against BUG

1.1 0th: when reading

审题! esp. input & output

1. the order of input variables
2. 注意数据是否为整数 [另: 写 ceil 时 - ceil(n/2) 是错的]
   - variant: 注意数据是否排好序
3. the output format
4. **data range. 2^{25} or 10^{25}?**
5. **special case:** n=0, n=1 是是否需要特判?
   - 1072C: n=0
   - 1079G: n=1

1.2 1st: when coding

1. draft
   - ? 算法流程, 有几个模块
   - ? 初始化, 数组边界, 下列情况必须特判边界:
     - 带优化的 dp
     - 多次使用的 dp
   - ? 清晰, **一致**, 映射关系完整的表达式 (核心步骤)：包括不限于: dp 方程, 坐标关系, ds 维护, 数学表达式
   - ! special case, 无解情况的判断
2. template
3. focus

1.3 2nd: g++ warning

```
g++ -Wall -Wextra -g3 -DLOCAL -Wshadow -Wpointer-arith -Wcast-qual -Winline -Wunreachable-code
```

1.4 3rd: static debug

1. not AC?
   - RE? 首先考虑数组 overflow 和 **0 除法** (1138D)
   - 小范围数组 overflow 只会 WA 而不会 RE
   - 强制在线可以将所有 WA 变成 RE
   - 大数据下 WA 考虑乘法溢出
   - 某些数据结构写错可能会导致 tle 而不会 wa
   - MLE? 256mb 下, ll[] 的极限是 3e7
     - 注意点的编号影响数组大小
     - 留意某些空间复杂度并不单调: 如 sqrt(N/logN) 分块
2. WA 的常见方向
   (a) overflow
   - 乘法溢出
     - 某个表达式忘记取模 (连乘 3 次忘记取模—)
     - 尽量用除法而不是乘法 (大数据时尤其小心乘法)：e.g. 统计 n! 中 p 的幂次
     - 用 a*b>0 判断是否有 0 时, 溢出… (可以用 a*b!=0) (1132A)
   - 数据类型错误:
     - 指标量 n, m 参与运算
     - 中间结果保存成 int (cur, cnt, stk[])
     - 忘记把 bool 改成 int
   - **rep long long 导致溢出**
   - ? INF 不够大

   (b) 错误的映射关系 (草稿不完善)
   - 直接使用循环变量而忽略数组，常出现于:
     - 大数据结构 e.g. 虚树 vtx, 树剖 loc
     - 某种变换后 e.g. 排序后, 离散化后
   - 默认根为 1; 根可否为叶子? 双连通分量要判断根节点;

   (c) 浮点陷阱
   - 四舍五入时，输出负 0
   - 0 * inf 导致 nan (斜率优化时, 分母为 0 特判)
3. static debug
   (a) 用 gedit light theme
   (b) 关注:
   - 表达式, 符号, 优先级 (三目运算符加括号)
   - 变量名错位, 映射关系
   - for 范围, 端点 (字符串, 多项式), rep 和 per
   - 数据类型, overflow

1.5 4th: gdb & sample testing

1.5.1 以下情况造数据测试:

1. 离散化
2. 多组数据
3. 存在二维对称量 (m 和 n)
4. sample 非常弱
5. 易 overflow 的数学题 (+ 编译选项 fsanitize=undefined)