# 恢复合影

有 n 个孩子,他们的编号为 1 ~ n,老师将他们根据身高排成一条直线(注意身高和编号的顺序无关),打算拍一张集体照。

但他们非常淘气,接下来的故事连续发生了五次：

1、老师将孩子们带到了正确的位置, 然后离开摆弄相机

2、一部分孩子乘机开小差,胡乱地插到其他孩子的位置中间

3、老师按下快门,得到一张照片,发现有些孩子位置不对

4、老师将所有开小差的孩子教育了一顿,被教育的孩子接下来就不会淘气,也不会乱跑了,返回第一步

注意,每一轮的流程中,都可能有新的孩子出来捣乱

你能从这五张照片里,还原出老师心目中原本正确的排位么？

这题保证1 ≤ n ≤ 100000

在正确的排列中的任意两个人x和y，假设x在y的左侧

在这五次照片中，x最多只有2次可能在y的右侧 1.x跑到y的右侧 2.y跑到x的左侧

其余都是x在y的左侧，因此这5次照片中超过3次出现x在y的左侧，那正确顺序x一定在y的左侧

核心代码:

bool cmp(int *x*, int *y*) {

    int left = 0; // 记录5次照片中x出现在y左侧的次数

    for (int i = 1; i <= 5; i++)

        if (pos[*x*][i] < pos[*y*][i]) left++;

    return left >= 3; // 大于三次, 一定是正确的顺序

}

# 逆序对翻转问题

给一个1~n的排列, m次询问,每次询问区间前反转这个区间  
如： {1,2,3}→{3,2,1}

再询问问这个排列中逆序对数的奇偶

*注意前面反转的区间是会影响之后的反转的,是强制在线的题目*

核心代码:

LL merge\_sort(int *q*[], int *l*, int *r*) {

    if (*l* >= *r*) return 0;

    int mid = (*l* + *r*) >> 1;

    LL res = merge\_sort(*q*, *l*, mid) + merge\_sort(*q*, mid + 1, *r*);

    int k = 0, i = *l*, j = mid + 1;

    while (i <= mid && j <= *r*)

        if (*q*[i] <= *q*[j]) tmp[k++] = *q*[i++];

        else {

            res += mid - i + 1;

            tmp[k++] = *q*[j++];

        }

    while (i <= mid) tmp[k++] = *q*[i++];

    while (j <= *r*) tmp[k++] = *q*[j++];

    for (i = *l*, j = 0; i <= *r*; i++, j++) *q*[i] = tmp[j];

    return res;

}