

Codeforces Round #590 (Div. 3) 翻译及题目解析

• 版权声明:

作者: littleseven

blog: <http://littleseven.top/>

QQ: 3364445435

此篇文章为原创文章, 未经允许不得转载, 侵权者必究。如有需要请联系QQ。

CF1234A Equalize Prices Again

• 题目翻译:

你是附近一家小超市的店长兼管理员。在你的超市中一共有 n 个物品, 第 i 个物品售价 a_i 元。

由于顾客经常会向你询问各种商品的价格, 所以你需要把所有价格都背下来, 这使你很苦恼。你准备使自己的生活更加简单而美好, 你准备为这 n 种商品定一个相同的价钱。

然而, 作为一个商店店长, 你一定不希望在价格调整过程中有任何亏损, 所以你想要选择一个价格使得这 n 种商品的总价格要大于等于所有商品原价的总和。也就是说按照调整的价格卖出所有商品的收入要大于等于原来价格卖出所有商品的总收入。

此外, 你也不想因为过高的价格使自己的顾客流失, 所以你需要找到所有满足条件的价格的最小值。

因此, 你需要找到对于这 n 种商品的最低可能均等价格, 并且按照这个价格出售所有商品的收入要与以原价出售所有商品的获利相等或者更多。

每个测试点有 q 组测试数据。

简要题意:

给出一个整数 n , 和 n 个整数 $a_1 - a_n$, 求出一个数 k 使得 $n * k \geq \sum_{i=1}^n (a_i)$ 且 k 最小。

每个测试点有 q 组测试数据。

• 题目解析:

其实很简单的一道题, 求所有商品价格平均值向上取整就可以了。

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

using namespace std;

#define eps 1e-5

char *p1, *p2, buf[100000];

#define nc() (p1 == p2 && (p2 = (p1 = buf) + fread(buf, 1, 100000, stdin), p1 == p2) ? EOF : *p1++)

int rd() {
    int x = 0, f = 1;
    char c = nc();
    while (c < 48) {
        if (c == '-') {
            f = -1;
        }
        c = nc();
    }
    while (c > 47) {

```

```

        x = (((x << 2) + x) << 1) + (c ^ 48);
        c = nc();
    }
    return x * f;
}

int q, n;

int main() {
    q = rd();
    while (q -- ) {
        int n = rd(), ans = 0;
        for (int i = 1; i <= n; i ++ ) {
            int a = rd();
            ans += a;
        }
        if (ans % n == 0) {
            printf("%d\n", ans / n);
        }
        else {
            printf("%d\n", ans / n + 1);
        }
    }
    return 0;
}

```

CF1234B1 Social Network (easy version)

• 题目翻译：

注：简单版与困难版的区别只是 n 和 k 的数据约束不同。

你正在通过你的手机在一个流行社交网络（朋友圈）之中发信息。你的手机可以同时展示至多 k 个你与你的朋友的最近通话记录。起初，手机的屏幕上空的（即显示的通话记录数量为0）。

每次通话都是你与你的朋友进行的。而且你与你的任何一个朋友至多只会通话一次，所以每次通话都是和你的朋友——对应、唯一定义的。

你（突然之间!!!）得到了预知未来的能力。你知道在一天当中你将会收到 n 条信息，第 i 条信息将会是来自编号为 id_i 的朋友的（ $1 \leq id_i \leq 10^9$ ）。

如果你收到的一条来自 id_i 的朋友的信息并且这个朋友已经在你手机屏幕所显示的通话记录中存在于，那么什么都不会发生：屏幕上的记录不会改变，顺序也不会被改变，你只是读了这个信息之后等待一条新的信息。

然而（在屏幕上没有显示与发来信息的朋友 id_i 的通话记录）：

- 首先，如果屏幕上显示的通话记录的数目为 k ，那么将在屏幕中清除最后一个通话记录（位置为 k ）。
- 现在屏幕上的记录数目一定小于 k 并且编号为 id_i 的朋友也没有出现在你手机屏幕上的通话记录当中。
- 你与编号为 id_i 的朋友的通话记录将出现在屏幕上列表的第一个（最上边）的位置，其他的通话记录都将向后移动一个位置。

你的任务就是在处理完这所有的 n 条信息之后找到这时通话记录的列表（按照他们在屏幕上显示的顺序）。

简要题意：

模拟一个通话记录列表，你将收到 n 条信息，第 i 条信息来自你的朋友 id_i ，对于每个信息有如下操作：

- 如果列表中有 id_i ，那么什么都不做。
- 如果列表中没有 id_i ，那么就删除列表中最下边的一个记录，并且把其他所有记录下移一位，再把 id_i 放在最上边。

询问处理完 n 条信息之后屏幕上的通话记录，按照在屏幕上显示的顺序。

• 题目解析：

这道题我们涉及到如何维护这个列表的问题，那么我们先看这个列表的一些性质：

- 唯一：列表中的所有元素只能出现一次，列表中没有重复的元素。
- 顺序：列表中的所有元素按照都按照出现的顺序排序，并且每次修改要删掉最后一个并且后推。

根据这道题所模拟的过程，我们很自然的就会想到队列。但是由于这道题既有队头入队，又有队尾出队，那么我们选择双端队列（deque）。

但是另外的一个问题，如何保证这个列表中的唯一性（不重复性）。这怎么实现？？？——集合啊。

按照集合的不重复性来维护这个不重复的列表刚刚好。（set里找不到就是没有出现过的）

那么我们就用deque和set混合就好了。

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

using namespace std;

char *p1, *p2, buf[100000];

#define nc() (p1 == p2 && (p2 = (p1 = buf) + fread(buf, 1, 100000, stdin), p1 == p2) ? EOF : *p1++)

int rd() {
    int x = 0, f = 1;
    char c = nc();
    while (c < 48) {
        if (c == '-') {
            f = -1;
        }
        c = nc();
    }
    while (c > 47) {
        x = (((x << 2) + x) << 1) + (c ^ 48);
        c = nc();
    }
    return x * f;
}

const int N = 2e5 + 10;

int n, k;

bool vis[N];

int main() {
    n = rd(), k = rd();
    deque<int> q;
    set<int> s;
    for (int i = 1; i <= n; i++) {
        int x = rd();
        if (s.find(x) == s.end()) {
            q.push_front(x);
            s.insert(x);
            if (q.size() > k) {
                int now = q.back();
                s.erase(now);
                q.pop_back();
            }
        }
    }
    printf("%d\n", q.size());
    while (!q.empty()) {
        printf("%d ", q.front());
        q.pop_front();
    }
    return 0;
}
```

CF1234B2 Social Network (hard version)

• 题目翻译：

同CF1234B1

- 题目解析:

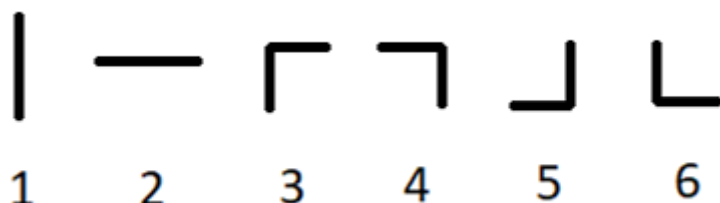
同CF1234B1

CF1234C Pipes

- 题目翻译:

参考: <https://www.luogu.org/problem/CF1234C>

给定一些特定形状的水管。有一个网格，它包含两行，每行包含 n 条水管。左上角的水管的坐标为 $(1, 1)$ ，而右下角的水管的坐标为 $(2, n)$ 。有六种水管：两种是直的，四种是弯曲的。形状如图所示。

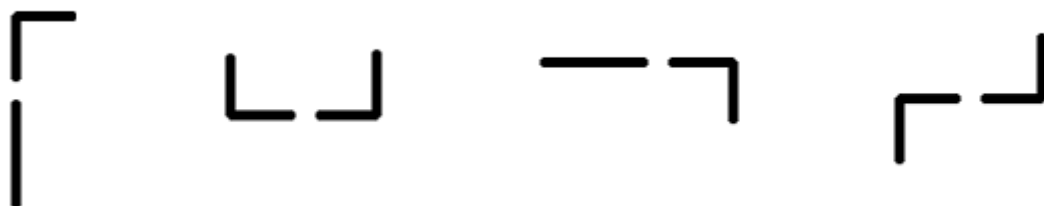


你可以任意地旋转水管（每次 90° ）。那么1, 2号水管，3, 4, 5, 6号水管就可以相互转换了。

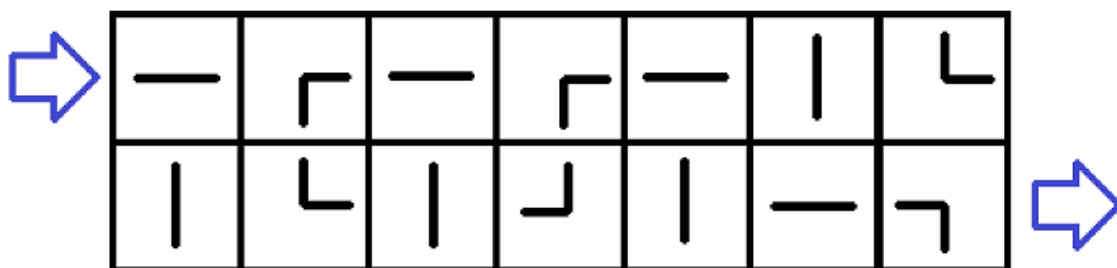
你想要令水流通过这些水管从 $(1, 0)$ (水流会从 $(1, 0)$ 流到左上角的水管)，流到 $(1, 1)$ ，再顺着连接着的水管到 $(2, n)$ 并向右流到 $(2, n+1)$ 。

当水管相邻并且当它们的末端均被连接，我们就称这些水管是连通的。

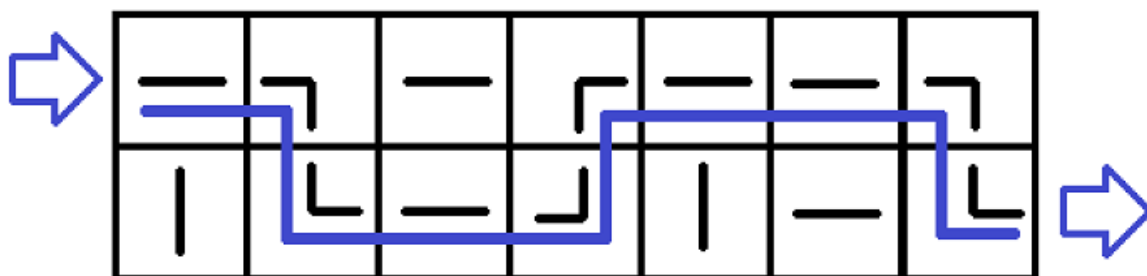
下面是连通水管的一些例子。



下面，我们将用一些例子说明这个问题。



它的解决方法如下图所示。



水流已经用蓝色的线标注了。要达到这个答案，我们要顺时针旋转 $(1, 2)$ 上的水管， $(2, 3)$ 上的水管， $(1, 6)$ 上的水管 90° ， $(1, 7)$ 及 $(2, 7)$ 上的水管 180° 。那么水流就可以从 $(1, 0)$ 到达 $(2, n+1)$ 。

你需要回答 q 次独立的询问。

简要题意：

现在有六种水管类型。

给出你两排水管的排列方式，判断水流是否可以从左上角流到右下角，可以输出 YES ，否则输出 NO 。

● 题目解析：

由于我们只有两行水管，那么我们直接模拟就好啊（我是一条小水流 QwQ ）。

我们可以发现一些有趣的性质（假设当前点为 (i, j) ， i, k 表示两行）：

- 如果 $(i, j + 1)$ 是直管(1, 2号)，那么我们只能走到 $(i, j + 1)$ 。
- 如果 $(i, j + 1)$ 和 $(k, j + 1)$ 都是弯管（3, 4, 5, 6号），那么我们只能走到 $(k, j + 1)$ 。

按照这个性质递归模拟就行，找到就是 YES ，否则就是 NO 。

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

using namespace std;

const int N = 2e5 + 10;

char s[N];

int mp[N][2];

int q, n, tag;

void work(int x, int y) {
    if (x > n) {
        return ;
    }
    if (x == n && y == 1) {
        tag = 1;
        return ;
    }
    if (mp[x + 1][y] == 0) {
        work(x + 1, y);
    }
    else if (mp[x + 1][y] && mp[x + 1][y ^ 1]) {
        work(x + 1, y ^ 1);
    }
    return ;
}

int main() {
    scanf("%d", &q);
    while (q -- ) {
        tag = 0;
        scanf("%d", &n);
        scanf("%s", s + 1);
        for (int i = 1; i <= n; i ++ ) {
            if (s[i] == '1' || s[i] == '2') {
                mp[i][0] = 0;
            }
            else {
                mp[i][0] = 1;
            }
        }
        scanf("%s", s + 1);
        for (int i = 1; i <= n; i ++ ) {
            if (s[i] == '1' || s[i] == '2') {
                mp[i][1] = 0;
            }
            else {
                mp[i][1] = 1;
            }
        }
    }
}
```

```

    }
}
work(0, 0);
if (tag == 1) {
    printf("YES\n");
}
else {
    printf("NO\n");
}
}
return 0;
}

```

CF1234D Distinct Characters Queries

• 题目翻译：

现在给你一个由小写字母组成的字符串 s 和关于这个字符串的 q 次操作。

我们用字符串 $s[l;r]$ 表示字符串 $s_l s_{l+1} \dots s_r$ 。举例来说, "codeforce"的字串有"code", "force", "f", "for", 但是没有"coder", "top"。

下面是两个类型的操作：

- 1 $pos\ c$ ($1 \leq pos \leq |s|$, c 是一个小写字母) 将 s_{pos} 位置的字符换成 c 。
- 2 $l\ r$ ($1 \leq l \leq r \leq |s|$) 计算在子串 $s[l,r]$ 当中有几种不同的字母。

简要题意：

同题目翻译。

• 题目解析：

看着给出的这两类型的操作这么熟呢???

线段树

树状数组

那就整啊!!!!

26颗线段树 OR 26颗树状数组

(显然后者好写, 那就写)

数据结构题直接看代码就好, 这是26颗树状数组。

```

#include <bits/stdc++.h>

using namespace std;

int tree[26][100010];

int n, m;

char s[100010];

void update(int ch, int x, int v) {
    for (int i = x; i <= n; i += (i & -i)) {
        tree[ch][i] += v;
    }
}

int query(int ch, int x) {
    int res = 0;

```

```

        for (int i = x; i; i -= (i & -i)) {
            res += tree[ch][i];
        }
        return res;
    }

    int main() {
        scanf("%s", s + 1);
        n = strlen(s + 1);
        for (int i = 1; i <= n; i++) {
            update(s[i] - 'a', i, 1);
        }
        scanf("%d", &m);
        for (int i = 1; i <= m; i++) {
            int opt;
            scanf("%d", &opt);
            if (opt == 1) {
                int a;
                char ch;
                scanf("%d %c", &a, &ch);
                update(s[a] - 'a', a, -1);
                update(ch - 'a', a, 1);
                s[a] = ch;
            }
            else {
                int a, b, ans = 0;
                scanf("%d%d", &a, &b);
                for (int i = 0; i < 26; i++) {
                    if ((query(i, b) - query(i, a - 1))) {
                        ans++;
                    }
                }
                printf("%d\n", ans);
            }
        }
        return 0;
    }
}

```

CF1234E Special Permutations

• 题目翻译：

我们定义 $p_i(n)$ 表示一个序列 $[i, 1, 2, \dots, i-1, i+1, \dots, n]$ 。即第 i 个序列表示把 i 放在序列首，其他正常。

举例：

- $p_1(4) = [1, 2, 3, 4]$;
- $p_2(4) = [2, 1, 3, 4]$;
- $p_3(4) = [3, 1, 2, 4]$;
- $p_4(n) = [4, 1, 2, 3]$;

现在给你一个数组 x_1, x_2, \dots, x_m ($1 \leq x_i \leq n$)。

我们定义 $pos(p, val)$ 表示 val 在 p 当中的位置，所以说 $pos(p_1(4), 3) = 3, pos(p_2(4), 2) = 1, pos(p_4(4), 4) = 1$ 。

之后在定义 $f(p) = \sum_{i=1}^{m-1} |pos(p, x_i) - pos(p, x_{i+1})|$ ，在这里 $|val|$ 表示 val 的绝对值。这个函数表示在 p 中 x 的相邻元素之间距离之和。

现在你的任务就是计算 $f(p_1(n)), f(p_2(n)), f(p_3(n)), \dots, f(p_n(n))$ 的值。