

STL综合练习

人员

温郝冬、柳力玮、谢梓轩、李知朔、韩昱辰、赵牧之、田心一、李瑞涵、初锦阳 到课, 刘宸熙、苑钊、王馨琪、刘子轩 线上

上周作业检查

上周作业链接: <https://www.luogu.com.cn/contest/251034>

2025-0608周日15:30

报名

编辑比赛

题目数4 | 报名人数17

比赛说明

题目列表

排行榜

名次	参赛者	总分	A	B	C	D
#1	柳力玮	400 (2.09h)	100 (9.40s)	100 (199ms)	100 (3.21s)	100 (2.08h)
#2	赵牧之	400 (2.48d)	100 (4.97s)	100 (204ms)	100 (1.23d)	100 (1.25d)
#3	田心一	400 (13.66d)	100 (7.17s)	100 (201ms)	100 (6.81d)	100 (6.85d)
#4	谢梓轩	400 (13.75d)	100 (5.93s)	100 (239ms)	100 (6.86d)	100 (6.89d)
#5	初锦阳	400 (27.80d)	100 (6.92d)	100 (6.94d)	100 (6.96d)	100 (6.98d)
#6	苑钊	390 (13.96d)	100 (7.63s)	100 (215ms)	100 (6.99d)	90 (6.97d)
#7	李知朔	310 (12.19d)	100 (7.08s)	100 (199ms)	100 (6.09d)	10 (6.11d)
#8	韩昱辰	300 (4.23h)	100 (2.08h)	100 (201ms)	100 (2.15h)	
#9	李瑞涵	200 (5.81s)	100 (5.61s)	100 (197ms)		
#10	温郝冬	200 (5.92s)	100 (5.72s)	100 (198ms)	0	
#11	纪博涵	200 (7.47s)	100 (7.26s)	100 (216ms)		
#12	王馨琪	200 (7.51s)	100 (7.28s)	100 (239ms)	0	
#13	刘派	200 (13.52d)	100 (6.75d)	100 (6.77d)	0	
#14	刘宸熙	154 (208ms)	54 (0ms)	100 (208ms)		0
#15	姜皓轩	100 (5.06s)	100 (5.06s)			

您正在共享屏幕 结束共享

作业

<https://www.luogu.com.cn/contest/252014> (课上讲了 A ~ E 题, 课后作业是 F 题, 其中 E 题比较难一些, 不要求同学们必须掌握)

课堂表现

今天主要讲了 deque 数组 和 map 套 deque 的应用, 这些内容不难, 就是第一次接触这些内容会有些不好理解, 同学们课下要着重复习一下第二题和第三题。

课堂内容

P1540 [NOIP 2010 提高组] 机器翻译

每次放一个数进队列前, 先扫一遍队列, 判断里面有没有这个数即可

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

using namespace std;

int main()
{
    deque<int> dq;
    int m, n, res = 0; cin >> m >> n;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        bool flag = false;
        for (int j : dq) {
            if (j == x) flag = true;
        }
        if (!flag) {
            ++res;
            dq.push_back(x);
            if (dq.size() > m) {
                dq.pop_front();
            }
        }
    }
    cout << res << endl;
    return 0;
}
```

U571068 机器翻译2

用一个 bool 数组, 表示一个数在不在队列中, 这样就不用每次 $O(n)$ 扫一遍队列, 可以直接 $O(1)$ 判断了

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

using namespace std;

const int maxn = 5e5 + 5;
bool st[maxn];

int main()
{
    queue<int> q;
    int m, n, res = 0; cin >> m >> n;
```

```

while (n -- ) {
    int x; cin >> x;
    if (st[x]) continue;

    ++res;
    q.push(x); st[x] = true;
    if ((int)q.size() > m) {
        int t = q.front(); q.pop(); st[t] = false;
    }
}

cout << res << endl;
return 0;
}

```

U499172 找每个数出现的第 k 个位置

用一个 deque 数组, dq[i] 里面存 i 这个数分别在哪些位置出现过

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 5e5 + 5;
deque<int> dq[maxn];

int main()
{
    int n, q; cin >> n >> q;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x; dq[x].push_back(i);
    }

    while (q -- ) {
        int x, k; cin >> x >> k;
        cout << dq[x][k-1] << endl;
    }
    return 0;
}

```

U571092 一个数出现的第 k 个位置

数据范围变大, 每个数可以是 $1e9$ 级别, 所以不能直接开 deque 数组, 可以用 `map<int, deque<int>>` 结构, 存储每个数分别在哪些位置出现过

```

#include <bits/stdc++.h>

using namespace std;

```

```

int main()
{
    map<int, deque<int>> mp;
    int n, T; cin >> n >> T;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x; mp[x].push_back(i);
    }

    while (T -- ) {
        int x, k; cin >> x >> k;
        if ((int)mp[x].size() < k) cout << -1 << endl;
        else cout << mp[x][k-1] << endl;
    }
    return 0;
}

```

P1478 陶陶摘苹果（升级版）

把所有苹果按照需要的力气排序一遍, 然后从前往后扫挑 需要力气小 的摘就可以了

```

#include <bits/stdc++.h>

using namespace std;

struct node {
    int x, y;
};
const int maxn = 5000 + 5;
node w[maxn];

bool cmp(node p, node q) {
    return p.y < q.y;
}

int main()
{
    int n, s; cin >> n >> s;
    int a, b; cin >> a >> b;
    for (int i = 1; i <= n; i++) {
        cin >> w[i].x >> w[i].y;
    }
    sort(w+1, w+n+1, cmp);

    int cnt = 0;
    for (int i = 1; i <= n; i++) {
        if (w[i].x <= a+b && s >= w[i].y) {
            cnt++;
            s -= w[i].y;
        }
    }
    cout << cnt << endl;
}

```

```
    return 0;
}
```

P1532 卡布列克圆舞曲

数 n 可以一直往下变, 直到变出一个以前出现过的数, 那么这个数就是循环节的起点。

从这个数作为起点再变一圈变回这个数, 就能找全循环节了。

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

using namespace std;

typedef long long LL;

LL calc(LL n) {
    vector<int> vec;
    while (n != 0) vec.push_back(n%10), n /= 10;

    sort(vec.begin(), vec.end());
    LL minn = 0;
    for (int i : vec) minn = minn*10 + i;

    reverse(vec.begin(), vec.end());
    LL maxx = 0;
    for (int i : vec) maxx = maxx*10 + i;

    return maxx - minn;
}

void solve(LL n) {
    set<LL> s;
    while (!s.count(n)) {
        s.insert(n); n = calc(n);
    }

    s.clear();
    while (!s.count(n)) {
        cout << n << " ";
        s.insert(n); n = calc(n);
    }
    cout << endl;
}

int main()
{
    LL n;
    while (cin >> n) solve(n);
    return 0;
}
```