

栈和队列

人员

初锦阳、王馨琪、刘宸熙、温郝冬、柳力玮、田心一、姜皓轩、谢梓轩、李知朔、韩昱辰、燕润石、李瑞涵、纪博涵、苑钊、赵牧之 到课, 栾婷婷、刘子轩 线上

上周作业检查

上周作业链接: <https://www.luogu.com.cn/contest/244895>

2025-0504周日15:30

报名

编辑比赛

题目数5 | 报名人数16

比赛说明

题目列表

排行榜

名次	参赛者	总分	A	B	C	D	E
#1	柳力玮	500 (4.22d)	100 (175ms)	100 (109ms)	100 (39ms)	100 (2.11d)	100 (2.12d)
#2	赵牧之	500 (17.76d)	100 (4.17d)	100 (118ms)	100 (38ms)	100 (6.78d)	100 (6.81d)
#3	田心一	500 (20.70d)	100 (6.91d)	100 (103ms)	100 (37ms)	100 (6.96d)	100 (6.83d)
#4	谢梓轩	448 (20.75d)	100 (6.91d)	100 (122ms)	100 (1.36s)	100 (6.91d)	48 (6.93d)
#5	栾婷婷	440 (20.64d)	100 (6.87d)	100 (107ms)	100 (38ms)	100 (6.87d)	40 (6.90d)
#6	王馨琪	324 (1.60d)		100 (132ms)	100 (37ms)	100 (18.91h)	24 (19.37h)
#7	刘宸熙	316 (13.95d)	50 (6.98d)	100 (105ms)	100 (38ms)	10 (0ms)	56 (6.96d)
#8	初锦阳	300 (3.14d)		100 (1.04d)	100 (1.03d)		100 (1.07d)
#9	李知朔	300 (5.21d)		100 (103ms)	100 (42ms)	100 (5.21d)	0
#10	燕润石	300 (20.86d)	100 (6.97d)	100 (6.95d)	100 (6.94d)		
#11	李瑞涵	210 (2.25h)		100 (121ms)	100 (40ms)	10 (2.25h)	
#12	姜皓轩	200 (3.14d)		100 (106ms)	100 (3.14d)		0
#13	温郝冬	110 (102ms)		100 (102ms)	10 (0ms)		
#14	刘子轩	100 (121ms)		100 (121ms)			
#15	韩昱辰	100 (19.50h)		100 (19.50h)			
#16	半仙胡小桃	100 (6.94d)			100 (6.94d)		

作业

<https://www.luogu.com.cn/contest/246120> (课上讲了 A ~ E 题, 课后作业是 F 题)

课堂表现

今天主要讲了 stack 和 queue 的内容, 同学们课上听讲都很认真, 也都学的很不错。

有几位同学课上没写出 A 题和 D 题, 课下需要再自习找一找自己代码的问题。

课堂内容

P3131 [USACO16JAN] Subsequences Summing to Sevens S

把所有前缀和取余一下 7, 找 0 1 2 ... 6 每个前缀和对应的最前面和最后面的位置即可。

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

using namespace std;

typedef long long LL;
const int maxn = 5e4 + 5;
int w[maxn];
LL pre[maxn];
int p[10], s[10];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    for (int i = 1; i <= n; ++i) {
        pre[i] = pre[i-1] + w[i]; pre[i] %= 7;
    }

    for (int i = 0; i <= 6; ++i) p[i] = s[i] = -1;
    for (int i = 0; i <= n; ++i) {
        int t = pre[i];
        if (p[t] == -1) p[t] = i;
    }
    for (int i = n; i >= 0; --i) {
        int t = pre[i];
        if (s[t] == -1) s[t] = i;
    }

    int res = 0;
    for (int i = 0; i <= 6; ++i) {
        int l = p[i], r = s[i];
        if (l == -1 || r == -1) continue;
        res = max(res, r-l);
    }
    cout << res << endl;
    return 0;
}
```

栈和队列的使用方法

```
#include <stack>
stack<int> s;
```

```
// 往栈里面插入数
s.push(1);
// 栈的大小
cout << s.size() << endl;
// 栈顶的数
cout << s.top() << endl;
// 删除栈顶的数
s.pop();
// 判断栈是不是空
if (s.empty()) {
    cout << "yes" << endl;
} else {
    cout << "no" << endl;
}
// 把整个栈清空
while (!s.empty()) {
    s.pop();
}

#include <queue>
queue<int> q;
// 往队尾里面插入数
q.push(1);
// 队列的大小
cout << q.size() << endl;
// 队头的数
cout << q.front() << endl;
// 删除队列前面的数
q.pop();
// 判断队列是不是空
if (q.empty()) {
    cout << "yes" << endl;
} else {
    cout << "no" << endl;
}
// 把整个栈清空
while (!q.empty()) {
    q.pop();
}
```

B3614 【模板】栈

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

using namespace std;

typedef unsigned long long ULL;

int main()
{
    int T; cin >> T;
```

```
while (T -- ) {
    stack<ULL> s;
    int n; cin >> n;
    while (n -- ) {
        string op; cin >> op;
        if (op == "pop") {
            if (s.empty()) cout << "Empty" << "\n";
            else s.pop();
        } else if (op == "query") {
            if (s.empty()) cout << "Anguei!" << "\n";
            else cout << s.top() << "\n";
        } else if (op == "size") {
            cout << s.size() << "\n";
        }
        else {
            ULL x; cin >> x; s.push(x);
        }
    }
}
return 0;
}
```

P1739 表达式括号匹配

从左到右扫一遍, 每次遇到左括号就用栈维护待匹配的左括号, 遇到右括号就找到栈里面的最后一个左括号进行匹配即可。

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

using namespace std;

int main()
{
    string str; cin >> str;
    stack<char> stk;
    for (char i : str) {
        if (i == '(') stk.push('(');
        else if (i == ')') {
            if (!stk.empty()) stk.pop();
            else { cout << "NO" << endl; return 0; }
        }
    }

    if (!stk.empty()) cout << "NO" << endl;
    else cout << "YES" << endl;
    return 0;
}
```

U516889 combine

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

using namespace std;

int main()
{
    stack<int> stk;
    int n; cin >> n;
    while (n -- ) {
        int x; cin >> x;
        stk.push(x);

        while ((int)stk.size() >= 2) {
            int a = stk.top(); stk.pop();
            int b = stk.top(); stk.pop();
            if (a == b) stk.push(a+1);
            else {
                stk.push(b), stk.push(a);
                break;
            }
        }
    }
    cout << stk.size() << endl;
    return 0;
}
```

B3616 【模板】队列

```
#include <iostream>
#include <queue>

using namespace std;

int main()
{
    queue<int> q;
    int n; cin >> n;
    while (n -- ) {
        int op; cin >> op;
        if (op == 1) {
            int x; cin >> x;
            q.push(x);
        } else if (op == 2) {
            if (!q.empty()) q.pop();
            else cout << "ERR_CANNOT_POP" << endl;
        } else if (op == 3) {
            if (!q.empty()) cout << q.front() << endl;
            else cout << "ERR_CANNOT_QUERY" << endl;
        } else {
            cout << q.size() << endl;
        }
    }
}
```

```
    }  
}  
return 0;  
}
```

P1996 约瑟夫问题

```
// 方法一，用一维数组进行模拟  
#include <bits/stdc++.h>  
  
using namespace std;  
  
bool f[105];  
  
int main()  
{  
    int n, k; cin >> n >> k;  
    int sum = n, id = 1, cnt = 0;  
    while (sum != 0) {  
        if (f[id] == false) {  
            cnt ++ ;  
        }  
        if (cnt % k == 0 && f[id] == false) {  
            f[id] = true;  
            cout << id << " ";  
            sum -- ;  
        }  
        id++;  
        if (id > n) {  
            id = 1;  
        }  
    }  
    cout << endl;  
    return 0;  
}
```

```
// 方法二，用 queue 进行模拟  
#include <bits/stdc++.h>  
  
using namespace std;  
  
int main()  
{  
    int n, m; cin >> n >> m;  
    queue<int> q;  
    for (int i = 1; i <= n; ++i) q.push(i);  
  
    int cnt = 0;  
    while (!q.empty()) {
```

```
    int t = q.front(); q.pop();  
    ++cnt;  
    if (cnt == m) {  
        cout << t << " ";  
        cnt = 0;  
    } else q.push(t);  
}  
cout << endl;  
return 0;  
}
```