

一维前缀和

人员

韩承睿、牟茗、辛帅辰、高健桓、李翰如、方俊喆、刘祺、夏硕承、秦显森、齐振玮、谢亚锴、王静嘉、牛同泽、徐浩然、齐振玮 到课，崔吉诺 线上

作业检查

谢亚锴、韩承睿 完成 E 题，牟茗、高健桓、辛帅辰 得了一部分分。其他同学未做 E 题。

课上题目 和 课下作业 有 4 位同学存在抄题解情况，已经在课上对这些同学进行点名批评，希望以后进行改正。

作业

<https://www.luogu.com.cn/contest/193009>

A、B、C、D 题

课堂表现

课堂纪律整体比较好，有几位同学可能很久没写代码忘记了，需要课下多写题复习。

课堂内容

T485179 corona

纯模拟题，非常适合大家在课下多写几遍，来提升自己代码能力。

这个题目一定要做到写熟。

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

using namespace std;

const int maxn = 100 + 5;
bool a[maxn][maxn];
int a_len[maxn];
int b[maxn];

int main() {
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a_len[i];
        for (int j = 1; j <= a_len[i]; j++) {
            int x; cin >> x;
            a[i][x] = true;
        }
    }
}
```

```

    }

    int x; cin >> x;
    int minn = 1e9;
    for (int i = 1; i <= n; i++) {
        if (a[i][x] == true) {
            minn = min(minn, a_len[i]);
        }
    }

    if (minn == 1e9) {
        cout << 0 << endl;
        return 0;
    }

    int cnt = 0;
    for (int i = 1; i <= n; i++) {
        if (a[i][x] == true && a_len[i] == minn) {
            cnt++;
            b[cnt] = i;
        }
    }

    cout << cnt << endl;
    for (int i = 1; i <= cnt; i++) {
        cout << b[i] << " ";
    }
    cout << endl;
    return 0;
}

```

一维前缀和

$a[1] \sim a[r]$ 的区间和 == $(a[1] \sim a[r] \text{ 的和}) - (a[1] \sim a[l-1] \text{ 的和})$

所以可以维护一个 p 数组代表 a 数组的前缀和, 其中 $p[i] = a[1] + a[2] + \dots + a[i]$

利用上述方式求 $p[i]$ 比较慢, 所以可以用 $p[i] = p[i-1] + a[i]$ 的方法, $O(1)$ 求 $p[i]$

$a[1] \sim a[r]$ 的区间和 == $p[r] - p[l-1]$

B3612 【深进1.例1】求区间和

前缀和模板题

```

#include <bits/stdc++.h>
using namespace std;
const int maxn = 1e5 + 5;
int a[maxn], p[maxn];

```

```
int main(){
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        p[i] = p[i-1] + a[i];
    }

    int m; cin >> m;
    while (m -- ) {
        int l, r; cin >> l >> r;
        cout << p[r] - p[l-1] << endl;
    }
    return 0;
}
```

P8772 [蓝桥杯 2022 省 A] 求和

前缀和模板题2, 需要提取公因数即可

```
#include <bits/stdc++.h>
using namespace std;

const int maxn = 2e5 + 5;
int a[maxn], p[maxn];

int get_sum(int l, int r) {
    return p[r] - p[l-1];
}

int main(){
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        p[i] = p[i-1] + a[i];
    }

    long long ans = 0;
    for (int i = 1; i <= n; i++) {
        ans += (long long)a[i] * get_sum(i+1, n);
    }
    cout << ans << endl;
    return 0;
}
```

二维前缀和

$(x1, y1) \sim (x2, y2)$ 的矩阵和: $p[x2][y2] - p[x1-1][y2] - p[x2][y1-1] + p[x1-1][y1-1]$

维护矩阵和的过程: $p[i][j] = p[i-1][j] + p[i][j-1] - p[i-1][j-1] + a[i][j]$

```
for (int i = 1; i <= n; i++) {  
    for (int j = 1; j <= m; j++) {  
        p[i][j] = p[i-1][j] + p[i][j-1] - p[i-1][j-1] + a[i][j];  
    }  
}
```

同学们可以课下自己画图，再理解一下二维前缀和，然后做 D 题