

# 综合练习

## 人员

洪晨栋、洪晨棋、崔宸赫、于家瑞、郭栩睿 到课

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/884>

### 2025-0927周六10:30 (并查集)

刷新

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	cuichenhe	崔宸赫	490	8767	100	100	100	90	100
2	hongchenqi	洪晨棋	420	11407	100	100	100	100	20
3	hongchendong	洪晨栋	400	4457	100	100	100	100	
4	taohuisheng	陶汇笙	400	12184	100	100	100	100	
5	guoxurui	郭栩睿	300	8855	100	100	100	0	
6	renxiangyu	任翔宇	300	10886	100	100	0	100	
7	yujiarui	于家瑞	200	10914	100		0	100	

## 本周作业

<https://cppoj.kids123code.com/contest/952> (课上讲了 A ~ D 这些题, 课后作业是 E 题)

## 课堂表现

同学们上课听讲做题都很认真, 就是做题的时候小错误会比较多, 老是遗忘细节, 这节课做题表现最好的同学是洪晨棋同学, 提出表扬!

## 课堂内容

### 集合 (上周作业)

先用 埃氏筛 找出所有的质数, 然后枚举比  $p$  大的质数  $i$ , 找到  $a\sim b$  中所有  $p$  的倍数, 将他们利用并查集合并。

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

using namespace std;

const int maxn = 1e5 + 5;
bool st[maxn];
int f[maxn];

int fFind(int x) {
    if (f[x] != x) f[x] = fFind(f[x]);
    return f[x];
}
```

```

int main()
{
    for (int i = 2; i < maxn; i++) {
        if (st[i]) continue;
        for (int j = i+i; j < maxn; j+=i) st[j] = true;
    }
    for (int i = 1; i < maxn; ++i) f[i] = i;

    int a, b, p; cin >> a >> b >> p;
    int cnt = b - a + 1;
    for (int i = p; i < maxn; ++i) {
        if (st[i]) continue;

        int j = i;
        while (j < a) j += i;

        for (int k = j+i; k <= b; k+=i) {
            if (fFind(j) != fFind(k)) {
                int pj = fFind(j), pk = fFind(k);
                f[pj] = pk, --cnt;
            }
        }
    }
    cout << cnt << endl;
    return 0;
}

```

### [SCOI2005] 繁忙的都市

最少选边的数量是  $n-1$

边的最大值最小, 可以二分一个 mid, 把  $\leq \text{mid}$  的边选中, 把这个边所连的两个点连通, 最后看是否所有点都连通了

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
int f[maxn];
struct node {
    int u, v, value;
    bool operator < (const node& p) const { return value < p.value; }
};
vector<node> vec;
int n, m;

int fFind(int x) {
    if (f[x] != x) f[x] = fFind(f[x]);
    return f[x];
}

```

```

}

bool check(int mid) {
    for (int i = 1; i <= n; ++i) f[i] = i;

    int cnt = n;
    for (node it : vec) {
        int u = it.u, v = it.v, value = it.value;
        if (value > mid) continue;

        int pu = fFind(u), pv = fFind(v);
        if (pu != pv) f[pu] = pv, --cnt;
    }

    return cnt==1;
}

int main()
{
    cin >> n >> m;
    while (m -- ) {
        int u, v, value; cin >> u >> v >> value;
        vec.push_back({u, v, value});
    }

    int l = 1, r = 10000;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (check(mid)) r = mid-1;
        else l = mid+1;
    }
    cout << n-1 << " " << l << endl;
    return 0;
}

```

## 计数间隔

枚举  $i$  作为右端点, 查询前面一共有多少个点的前缀和是  $p[i]-k$ , 这个查询 + 维护过程可以用 map 实现

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn];
LL p[maxn];

int main()
{
    int n; LL k; cin >> n >> k;

```

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

map<LL, int> mp; mp[0] = 1;
LL res = 0;
for (int i = 1; i <= n; ++i) {
    res += mp[p[i]-k];
    mp[p[i]]++;
}
cout << res << endl;
return 0;
}
```

## Prefix K-th Max

用小根堆 维护前 k 大的数

如果一个数大于小根堆里的最小值, 就把最小值删除, 把这个数插入小根堆中

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

using namespace std;

int main()
{
    int n, k; cin >> n >> k;
    priority_queue<int, vector<int>, greater<int>>>q;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        if (i <= k) {
            q.push(x);
            if (i == k) cout << q.top() << endl;
        }
        else {
            if (x > q.top()) {
                q.pop(); q.push(x);
            }
            cout << q.top() << endl;
        }
    }
    return 0;
}
```