

递归练习

人员

郭栩睿、邹忆航、陶汇笙、罗启宸、崔宸赫 到课

作业检查

上周作业链接: <https://www.luogu.com.cn/contest/250961>

2025-0607周六10:30

报名

编辑比赛

题目数5 | 报名人数9

比赛说明

题目列表

排行榜

| 名次 | 参赛者 | 总分 | A | B | C | D | E |
|----|-----|----------------|---------------|---------------|----------------|----------------|----------------|
| #1 | 洪晨栋 | 500 (1.99h) | 100 (15ms) | 100 (15ms) | 100 (311ms) | 100 (7ms) | 100 (1.99h) |
| #2 | 洪晨棋 | 500 (2.16h) | 100 (15ms) | 100 (17ms) | 100 (302ms) | 100 (7ms) | 100 (2.16h) |
| #3 | 陶汇笙 | 500 (3.41d) | 100 (15ms) | 100 (17ms) | 100 (310ms) | 100 (7ms) | 100 (3.41d) |
| #4 | 罗启宸 | 500 (6.40d) | 100 (15ms) | 100 (15ms) | 100 (323ms) | 100 (1.96h) | 100 (6.31d) |
| #5 | 郭栩睿 | 400 (357ms) | 100 (16ms) | 100 (17ms) | 100 (316ms) | 100 (8ms) | |
| #6 | 邹忆航 | 400 (6.45d) | 100 (15ms) | 100 (15ms) | 100 (306ms) | 100 (6.45d) | |
| #7 | 崔宸赫 | 300 (343ms) | 100 (15ms) | 100 (19ms) | 100 (309ms) | | |
| #8 | 宋吉相 | 300 (2.33h) | 100 (15ms) | 100 (15ms) | 100 (2.33h) | | |
| #9 | 马敬杰 | 200 (323ms) | 100 (15ms) | | 100 (308ms) | | |

作业

<https://www.luogu.com.cn/contest/252010> (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

今天的 A、B、C 几道题, 表面看起来区别比较大, 其实本质都是一样的题, 通过递归找出所有可能的情况, 然后判断这种情况是否合法即可。

同学们课下要好好复习一下这几道题, 着重学透这几道题之后, 以后碰到这种题就比较简单了。

课堂内容

U504494 并集与交集

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

```
using namespace std;

int main()
{
    int n, m; cin >> n >> m;
    set<int> s, s2, s3;
    while (n -- ) {
        int x; cin >> x; s.insert(x);
        s2.insert(x);
    }

    while (m -- ) {
        int x; cin >> x; s.insert(x);
        if (s2.count(x)) s3.insert(x);
    }

    for (int i : s) cout << i << " "; cout << endl;
    for (int i : s3) cout << i << " "; cout << endl;
    return 0;
}
```

B3622 枚举子集（递归实现指数型枚举）

```
#include<iostream>
using namespace std;
char s[105];
int x;
void dfs(int n){
    if(n==0){
        for(int i=x;i>=1;i--){
            cout<<s[i];
        }
        cout<<endl;
        return;
    }
    s[n]='N';
    dfs(n-1);
    s[n]='Y';
    dfs(n-1);
}
int main()
{
    cin>>x;
    dfs(x);
    return 0;
}
```

P1036 [NOIP 2002 普及组] 选数

考虑每个数选或者不选, 最后判断是否符合题意

```
// 方法一，比较慢一些
#include <bits/stdc++.h>

using namespace std;

const int maxn = 20 + 5;
int w[maxn];
bool st[maxn];
int n, k, ans = 0;

bool is_prime(int x) {
    if (x <= 1) return false;
    for (int i = 2; i*i <= x; i++) {
        if (x%i == 0) return false;
    }
    return true;
}

void dfs(int u) {
    if (u == n+1) {
        int cnt = 0, sum = 0;
        for (int i = 1; i <= n; i++) {
            if (st[i]) {
                cnt++, sum += w[i];
            }
        }
        if (cnt==k && is_prime(sum)) ans++;
        return;
    }

    st[u] = false;
    dfs(u+1);
    st[u] = true;
    dfs(u+1);
}

int main()
{
    cin >> n >> k;
    for (int i = 1; i <= n; i++) cin >> w[i];
    dfs(1);
    cout << ans << endl;
    return 0;
}
```

```
// 方法二，dfs的过程中同时维护 cnt 和 sum，这样不用最后再扫一遍了，更快一些
#include <bits/stdc++.h>

using namespace std;
```

```

const int maxn = 20 + 5;
int w[maxn];
int n, k, ans = 0;

bool is_prime(int x) {
    if (x <= 1) return false;
    for (int i = 2; i*i <= x; i++) {
        if (x%i == 0) return false;
    }
    return true;
}

void dfs(int u, int cnt, int sum) {
    if (u == n+1) {
        if (cnt==k && is_prime(sum)) ans++;
        return;
    }

    dfs(u+1, cnt, sum);
    dfs(u+1, cnt+1, sum+w[u]);
}

int main()
{
    cin >> n >> k;
    for (int i = 1; i <= n; i++) cin >> w[i];
    dfs(1, 0, 0);
    cout << ans << endl;
    return 0;
}

```

U523946 coverage

考虑每个集合选或不选的所有情况, 最后把选的集合挑出来判断是否符合题目要求。

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 10 + 5;
set<int> s[maxn];
int n, m, res = 0;
bool st[maxn];

void dfs(int u) {
    if (u == m+1) {
        // 判断选出来的集合, 能否凑出 1~n
        set<int> bingji;
        for (int i = 1; i <= m; i++) {
            if (st[i]) { // 选第 i 个集合, 把第 i 个集合里的数放到并集里
                for (int j : s[i]) bingji.insert(j);
            }
        }
    }
}

```

```
        }
    }
    bool flag = true;
    for (int i = 1; i <= n; i++) { // 判断并集里面是否 1~n 都有
        if (!bingji.count(i)) flag = false;
    }
    if (flag) res++; // 如果 1~n 都有, 就 res++
    return;
}

st[u] = false;
dfs(u+1);
st[u] = true;
dfs(u+1);
}

int main()
{
    cin >> n >> m;
    for (int i = 1; i <= m; i++) {
        int len; cin >> len;
        for (int j = 1; j <= len; j++) {
            int x; cin >> x;
            s[i].insert(x);
        }
    }
    dfs(1);
    cout << res << endl;
    return 0;
}
```