

# 综合混练

## 人员

隋钰涵、杨瑾硕、秦显森、刘闯速、孙靖轲、郭骐嘉、赵熙羽、牛同泽、于子珈、高健桓、武敬哲、庞滢、陈洛冉、谢亚锴 到课，董浩桢 线上

## 上周作业检查

<https://www.luogu.com.cn/contest/244894>

2025-0504周日10:30

报名

编辑比赛

题目数4 | 报名人数15

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C	D
#1	孙靖轲	400 (5.10d)	100 (5.44h)	100 (326ms)	100 (10.09h)	100 (4.45d)
#2	谢亚锴	400 (8.15d)	100 (588ms)	100 (310ms)	100 (1.20d)	100 (6.95d)
#3	秦显森	400 (8.58d)	100 (8.41h)	100 (315ms)	100 (6.99d)	100 (1.24d)
#4	赵熙羽	400 (10.64d)	100 (1.41d)	100 (1.43d)	100 (6.36d)	100 (1.44d)
#5	郭骐嘉	400 (12.92d)	100 (572ms)	100 (325ms)	100 (6.45d)	100 (6.48d)
#6	于子珈	396 (13.09d)	100 (672ms)	100 (374ms)	100 (6.08d)	96 (7.00d)
#7	程梓豪	300 (11.27h)	100 (4.38h)	100 (2.14h)	100 (4.75h)	
#8	高健桓	300 (6.92d)	100 (605ms)	100 (316ms)	100 (6.92d)	
#9	武敬哲	300 (12.29d)	100 (6.14d)	100 (333ms)	100 (6.15d)	
#10	牛同泽	300 (13.88d)	100 (6.92d)	100 (330ms)		100 (6.95d)
#11	杨瑾硕	200 (593ms)	100 (493ms)	100 (100ms)		
#12	隋钰涵	200 (13.87d)	100 (6.94d)	100 (6.93d)		
#13	刘闯速	100 (2.01h)	100 (2.01h)		0 (0ms)	
#14	庞滢	20 (0ms)	20 (0ms)			

## 作业

<https://www.luogu.com.cn/contest/246119> (课上讲了 A ~ C 题, 课后作业是 D 题)

## 课堂表现

今天的 A、B 2 道题比较复杂一些, 同学们课上整体写的不是很好, 课下要好好再复习复习。

## 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;
}
```

### P1025 [NOIP 2001 提高组] 数的划分

记忆化搜索, 对于 n 这个数, 分成 k 份, 最小值给 limit 开头时, 能有多少份。

此时 dfs(n,k,limit) 只需要调用 dfs(n-i,k-1,i) 即可 (i >= limit)

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

```

using namespace std;

int f[205][10][205];

int dfs(int n, int k, int limit) {
    if (n < limit) return 0;
    if (k == 1) return 1;
    if (f[n][k][limit]) return f[n][k][limit];

    int res = 0;
    for (int i = limit; i <= n; ++i) {
        res += dfs(n-i, k-1, i);
    }
    f[n][k][limit] = res;
    return f[n][k][limit];
}

int main()
{
    int n, k; cin >> n >> k;
    cout << dfs(n, k, 1) << endl;
    return 0;
}

```

### P1330 封锁阳光大学

对每个联通块进行 dfs 黑白染色的搜索, 在每个联通块中找出黑、白块中少的那个, 然后全加起来。

黑白染色: 一个点染黑时, 周围点全部染白; 一个点染白时, 周围点全部染黑

如果中间出现了冲突, 则输出 -1

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e4 + 5;
vector<int> vec[maxn];
int f[maxn];
int cnt0, cnt1;

bool dfs(int u, int col) {
    if (f[u] != -1) return f[u]==col;

    f[u] = col;
    if (col == 0) cnt0++;
    else cnt1++;

    for (int i : vec[u]) {
        if (!dfs(i, 1-col)) return false;
    }
}

```

```

    return true;
}

int main()
{
    memset(f, -1, sizeof(f));
    int n, m; cin >> n >> m;
    while (m -- ) {
        int u, v; cin >> u >> v;
        vec[u].push_back(v), vec[v].push_back(u);
    }

    int res = 0;
    for (int i = 1; i <= n; ++i) {
        if (f[i] == -1) {
            cnt0 = 0, cnt1 = 0;
            if (!dfs(i, 0)) {
                cout << "Impossible" << endl;
                return 0;
            } else res += min(cnt0, cnt1);
        }
    }
    cout << res << endl;
    return 0;
}

```

### P9050 [PA 2021] Sumy

把所有鱼进行排序, 然后二分找分界点即可

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;
int a[maxn], b[maxn];

bool check(int n, int mid) {
    LL sum = b[mid];
    for (int i = 1; i <= n; ++i) {
        if (i == mid) continue;
        if (b[i] < sum) sum += b[i];
        else return false;
    }
    return true;
}

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

```

```
for (int i = 1; i <= n; ++i) cin >> a[i];

memcpy(b, a, sizeof(b));
sort(b+1, b+n+1);
int l = 1, r = n;
while (l <= r) {
    int mid = (l+r) / 2;
    if (check(n, mid)) r = mid-1;
    else l = mid+1;
}

for (int i = 1; i <= n; ++i) {
    if (l<=n && a[i]>=b[l]) cout << "T";
    else cout << "N";
}
cout << endl;
return 0;
}
```