

约瑟夫问题 + 二进制位运算相关难题

人员

韩承睿、牟茗、辛帅辰、高健桓、李翰如、崔吉诺、刘祺、夏硕承、齐振玮、谢亚锴、王静嘉、牛同泽、徐浩然、赵熙羽 到课

作业检查

韩承睿 已完成

牟茗 已完成

辛帅辰 已完成

高健桓 未打卡

李翰如 未打卡

方俊喆 未打卡

崔吉诺 未打卡

刘祺 已完成

夏硕承 已完成

秦显森 未打卡

齐振玮 未打卡

谢亚锴 已完成

王静嘉 未打卡

牛同泽 已完成

徐浩然 已完成

赵熙羽 第一次进班

作业

<https://www.luogu.com.cn/contest/186760>

A、B、C 3道题

课堂表现

同学们课上听讲都比较认真，不过这节课内容有些难，同学们课下要好好进行思考复习。

课堂内容

noi 1.13 20:话题焦点人物

cnt 数组实现桶的功能，记录一个人被提及过多少次

bool 类型的 tiji 数组用来表示 谁提及过谁

```
#include <iostream>

using namespace std;

int cnt[105];
bool tiji[105][105];

int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        int u;
        cin >> u;
        int k;
        cin >> k;
        for (int j = 1; j <= k; j++) {
            int x;
            cin >> x;
            cnt[x]++;
            tiji[u][x] = true;
        }
    }

    int maxx = 0;
    for (int i = 1; i <= 100; i++) {
        if (cnt[i] > maxx) {
            maxx = cnt[i];
        }
    }

    for (int i = 1; i <= 100; i++) {
        if (cnt[i] == maxx) {
            cout << i << endl;
            for (int j = 1; j <= 100; j++) {
                if (tiji[j][i]) {
                    cout << j << " ";
                }
            }
            cout << endl;
            break;
        }
    }
    return 0;
}
```

noi 1.13 02:不吉利日期

```
#include <iostream>

using namespace std;

int a[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

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

    for (int i = 1; i <= 12; i++) {
        // 1 ~ i-1
        int sum = 0;
        for (int j = 1; j <= i-1; j++) {
            sum += a[j];
        }
        sum += 12;

        int t = (w + sum) % 7;
        if (t == 0) {
            t = 7;
        }
        if (t == 5) {
            cout << i << endl;
        }
    }

    return 0;
}
```

P1996 约瑟夫问题

n, m

1. 目前的下标: pos=1
2. 目前报数几个人: cnt=0
3. 一共出圈几个人: sum=0

```
while (true) {
    if (位置超过总人数) {
        重新回到第一个位置
    }
    if (pos 这个位置之前出圈过) {
        pos++;
        continue;
    }
    pos 这个位置的人报数
    if (报数 == m) {
```

```

        cout << ...;
        这个位置的人出圈，这个位置打标记
        报数变为 0
        出圈人数 +1
        if (出圈人数 == n) {
            break;
        }
    }
    pos++;
}

```

此题不提供具体代码，要求同学们课下思考实现

CF1903D1 Maximum And Queries (easy version)

- 看 $a[1] \sim a[n]$ 这些数的第 i 位是不是 1
 怎么看一个数 x 的第 i 位是 0 还是 1 $\rightarrow (x \gg i) \& 1$
- 如果第 i 位是 0 的话，要计算给他加一个多少才能把这一位变为 1
 如果一个数 x 的第 i 位是 0
 现在我们想把 x 的第 i 位变为 1
 最少给 x 加几 $\rightarrow 2^i - x \& (2^i - 1)$
- 统计总次数时，确保不要炸 `long long`，中间超过 k 之后直接 `break`

- n, q 输入，并输入 a 数组
- 循环 q 次，每一次，都应该 a 数组 拷贝到 b 数组中，之后 `cin >> k` (k 是 `long long` 类型)
- 从高位往低位，看数组中的每一个数
- 看 $b[j]$ 这个数的第 i 位是不是 1
- 如果是 0，我们想把这一位变为 1，加一个 $(1 \ll i) - b[j] \& (1 \ll i)$

```

long long s = 1;
long long ans = 0;
for (int i = 62; i >= 0; i--) { // 从高位往低位处理
    long long sum = 0;
    for (int j = 1; j <= n; j++) {
        if (b[j] 这个数二进制的第 i 位是 0 的话) {
            sum += 把 b[j] 这个数二进制第 i 位变为 1 的代价
            if (sum > k) {
                break;
            }
        }
    }
}

// 一共需要 sum 次，才能把所有数的第 i 位变为 1
if (sum <= k) {
    这里需要消耗 sum 次修改的代价
    答案保留第 i 位为 1 的答案
    for (int j = 1; j <= n; j++) {
        if (b[j] 这个数二进制的第 i 位是 0 的话) {
            把 b[j] 的这一位变为 1
        }
    }
}

```

```
    }  
    }  
}  
cout << ans;
```

此题不提供具体代码，要求同学们课下独立思考实现