

# 蛇数统计

## 人员

夏志赫、武敬哲、高健桓、程梓豪、牛同泽、隋钰涵、李青默、赵熙羽、秦显森 到课

## 作业检查

上周作业链接: <https://www.luogu.com.cn/contest/222488>

2024-1229周日10:30

报名

编辑比赛

题目数3 | 报名人数17

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C
#1	牟茗	300 (4.31h)	100 (55.53min)	100 (2.10h)	100 (1.29h)
#2	程梓豪	300 (9.76h)	100 (1.10h)	100 (2.20h)	100 (6.46h)
#3	武敬哲	300 (5.54d)	100 (1.01h)	100 (1.81h)	100 (5.42d)
#4	赵熙羽	300 (12.25d)	100 (3.47d)	100 (3.48d)	100 (5.30d)
#5	秦显森	275 (20.91d)	100 (6.94d)	100 (6.98d)	75 (7.00d)
#6	隋钰涵	250 (3.12h)	100 (56.45min)	100 (1.82h)	50 (21.92min)
#7	刘祺	250 (3.32h)	100 (1.06h)	100 (1.80h)	50 (27.80min)
#8	高健桓	250 (3.70h)	100 (1.04h)	100 (2.15h)	50 (30.23min)
#9	夏志赫	250 (7.10d)	100 (54.32min)	100 (1.79h)	50 (6.99d)
#10	牛同泽	200 (3.05h)	100 (1.01h)	100 (2.03h)	
#11	董浩桢	190 (2.34h)	100 (1.16h)	90 (1.18h)	
#12	方俊喆	100 (1.65h)	100 (1.65h)		
#13	齐振玮	100 (1.76h)	100 (1.76h)		

## 作业

<https://www.luogu.com.cn/contest/223457>

## 课堂表现

今天课上讲了 蛇数 这个题目, 这个题目比较复杂, 同学们课下一定要好好复习一下这道题目。

## 课堂内容

ATcoder387 C - Snake Numbers

```
typedef long long LL;

LL solve(int x) { // 统计所有长度为 x 的答案
    LL sum = 0;
    for (int i = 1; i <= 9; i++) {
        sum += w_pow(i, x-1);
    }
    return sum;
}

LL calc(LL x) { // 求 1 ~ x 中有几个蛇数
    if (x<10) return 0;

    1. 确定字符串的长度是 n
    应该先统计 长度是2,3,4,...,n-1 的所有答案
    for (int i = 2; i <= n-1; i++) {
        sum += solve(i);
    }

    2. 第二步考虑长度是 n
    首位 s0:
    for (int i = 1; i <= s0-1; i++) {
        sum += w_pow(i, n-1);
    }

    3. 第三步, 首位已经确定是s0, 从第 2 位考虑到第 n 位
    for (int i = 2; i <= n; i++) {
        if (s[i] >= s0) {
            sum += w_pow(s0, n-i+1);
            break;
        } else {
            sum += s[i] * w_pow(s0, n-i);
            if (i == n) sum++;
        }
    }
}
```

## P7870 「Wdoi-4」兔已着陆

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

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;
struct node {
    int l, r;
};
stack<node> stk;
```

```
LL get_sum(int l, int r) { return 1LL*(l+r)*(r-l+1)/2; }

LL calc(LL k) {
    LL res = 0;
    while (!stk.empty()) {
        node p = stk.top(); stk.pop();
        int l = p.l, r = p.r, len = p.r-p.l+1;
        if (len <= k) k -= len, res += get_sum(l, r);
        else {
            int ll = r - k + 1;
            res += get_sum(ll, r);
            stk.push({l, ll-1});
            break;
        }
    }
    return res;
}

int main()
{
    int n; cin >> n;
    while (n -- ) {
        int op; cin >> op;
        if (op == 1) {
            int l, r; cin >> l >> r;
            stk.push({l, r});
        } else {
            LL k; cin >> k;
            cout << calc(k) << endl;
        }
    }
    return 0;
}
```

## P1007 独木桥

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

using namespace std;

int main()
{
    int L, n; cin >> L >> n;
    int res1 = 0, res2 = 0;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        res1 = max(res1, min(x, L+1-x));
        res2 = max(res2, max(x, L+1-x));
    }

    cout << res1 << " " << res2 << endl;
}
```

```
    return 0;  
}
```

### P1478 陶陶摘苹果（升级版）

按照 每个苹果需要的力气值 排序, 从小的力气值开始选即可。

注意, 当你目前的力气小于某个苹果需要的力气时, 不能选这个苹果。