

代价

人员

洪晨栋、洪晨棋、郭栩睿、宋吉相、陶汇笙、崔宸赫、王恩泽 到课

作业检查

上周作业链接: <https://www.luogu.com.cn/contest/243752>

2025-0426周六10:30

报名

编辑比赛

题目数4 | 报名人数14

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C	D
#1	陶汇笙	400 (11.74d)	100 (36ms)	100 (3.41d)	100 (121ms)	100 (8.33d)
#2	洪晨栋	400 (16.98d)	100 (37ms)	100 (8.36d)	100 (4.19h)	100 (8.44d)
#3	洪晨棋	400 (17.13d)	100 (37ms)	100 (8.46d)	100 (4.20h)	100 (8.50d)
#4	宋吉相	300 (4.86d)	100 (34ms)	100 (2.43d)	100 (2.42d)	
#5	郭栩睿	300 (6.43d)	100 (50ms)	100 (6.43d)	100 (110ms)	
#6	邹忆航	200 (149ms)	100 (39ms)		100 (110ms)	
#7	罗启宸	200 (6.49d)	100 (38ms)		100 (6.49d)	
#8	马敬杰	100 (39ms)	100 (39ms)			
#9	崔宸赫	100 (111ms)			100 (111ms)	
#10	王恩泽	100 (122ms)			100 (122ms)	

作业

<https://www.luogu.com.cn/contest/245142> (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

同学们课上听课做题表现都很认真。

课堂内容

T504032 克隆机 (clone)

先是 1 遍每个字符, 再是 2 遍每个字符, 再 4 遍, 再 8 遍, 16 遍,

嵌套循环维护即可, 要维护已经遍历过多少字母了, 看什么时候遍历能超过第 n 个字符即可

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

int main()
{
    int k; LL n; cin >> k >> n;
    LL len = 0;
    for (LL i = 1; ; i *= 2) {
        for (char j = 'A'; j <= 'A'+k-1; ++j) {
            len += i;
            if (len >= n) {
                cout << j << endl;
                return 0;
            }
        }
    }
    return 0;
}

```

P2708 硬币翻转

跟连胜统计一样, 统计有几个段

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e6 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);

    int res = 0;
    for (int i = 2; i <= n; ++i) {
        if (s[i] != s[i-1]) res++;
    }
    if (s[n] == '0') res++;

    cout << res << endl;
    return 0;
}

```

P3056 [USACO12NOV] Clumsy Cows S

用栈维护括号匹配的过程即可

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

using namespace std;

const int maxn = 1e5 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1), res = 0;
    stack<char> stk;
    for (int i = 1; i <= n; ++i) {
        char x = s[i];
        if (x == '(') stk.push(x);
        else {
            if (!stk.empty()) stk.pop();
            else {
                stk.push('('), ++res;
            }
        }
    }
    res += stk.size()/2;
    cout << res << endl;
    return 0;
}
```

U513572 代价 (cost)

从 1~n 枚举最终变成哪一个, 然后利用前缀和可以 $O(1)$ 求全变成 $w[i]$ 时的代价

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
LL w[maxn], p[maxn];

LL get_sum(int l, int r) {
    if (l <= r) return p[r] - p[l-1];
    return 0;
}

int main()
{
    int n, A, B; cin >> n >> A >> B;
    for (int i = 1; i <= n; ++i) cin >> w[i];
```

```
sort(w+1, w+n+1);
for (int i = 1; i <= n; ++i) p[i] = p[i-1] + w[i];

LL res = 4e18;
for (int i = 1; i <= n; ++i) {
    // 把 w[1]~w[i-1] 全变为 w[i] 的值, w[i+1]~w[n] 全变为 w[i] 的值
    LL pre_sum = get_sum(1, i-1), tar_pre_sum = (LL)w[i]*(i-1);
    LL suf_sum = get_sum(i+1, n), tar_suf_sum = (LL)w[i]*(n-i);

    LL diff_a = tar_pre_sum - pre_sum, diff_b = suf_sum - tar_suf_sum;
    res = min(res, diff_a*A + diff_b*B);
}
cout << res << endl;
return 0;
}
```