

# k 倍区间

## 人员

初锦阳、赵牧之、王馨琪、刘宸熙、倪炜艺、柳力玮、田心一、姜皓轩、谢梓轩、李知朔、韩昱辰、栾婷婷、纪博涵 到课

## 上周作业检查

上周作业链接: <https://www.luogu.com.cn/contest/238527>

2025-0330周日15:30

报名

编辑比赛

题目数6 | 报名人数21

比赛说明

题目列表

排行榜

名次	参赛者	总分	A	B	C	D	E	F
#1	刘宸熙	600 (29.60d)	100 (2.23d)	100 (1.58s)	100 (6.71d)	100 (6.72d)	100 (6.97d)	100 (6.96d)
#2	柳力玮	500 (2.19h)	100 (138ms)	100 (1.79s)	100 (233ms)	100 (66ms)	100 (2.19h)	
#3	初锦阳	500 (1.84d)	100 (4.39h)	100 (1.59s)	100 (5.11h)	100 (4.98h)	100 (1.24d)	0
#4	田心一	420 (20.50d)	100 (129ms)	100 (6.79d)	100 (232ms)	100 (6.80d)	20 (6.91d)	
#5	李知朔	410 (10.51d)	100 (2.14h)	100 (1.51s)	100 (2.20h)	100 (5.14d)	10 (5.19d)	
#6	赵牧之	400 (6.58d)	100 (122ms)	100 (1.68s)	100 (3.28d)	100 (3.29d)		
#7	栾婷婷	360 (12.58d)	100 (119ms)	100 (1.80s)	0 (0ms)	60 (6.26d)	100 (6.32d)	
#8	谢梓轩	340 (20.92d)	80 (0ms)	0 (0ms)	100 (6.97d)	60 (6.97d)	100 (6.98d)	
#9	苑钊	300 (2.27h)	100 (122ms)	100 (1.60s)	100 (2.27h)	0		
#10	姜皓轩	300 (4.20d)	100 (125ms)	100 (1.95s)	0 (0ms)	100 (4.20d)		
#11	倪炜艺	300 (6.83d)		100 (2.29s)	100 (263ms)		100 (6.83d)	
#12	韩昱辰	230 (2.19h)	100 (122ms)	30 (0ms)	100 (2.19h)	0		
#13	刘子轩	225 (1.58s)	40 (0ms)	100 (1.58s)	85 (0ms)			
#14	温郝冬	200 (1.76s)	100 (122ms)	100 (1.64s)				
#15	王馨琪	200 (5.58h)	100 (5.58h)	100 (1.51s)	0			
#16	刘梓勋	200 (14.14d)			100 (7.07d)	100 (7.07d)		
#17	周沁言	200 (14.14d)			100 (7.06d)	100 (7.08d)		
#18	燕润石	140 (2.14s)	40 (0ms)	100 (2.14s)				
#19	李瑞涵	30 (0ms)		30 (0ms)				
#20	宋吉相	0 (0ms)	0 (0ms)					

## 作业

<https://www.luogu.com.cn/contest/240403> (课上讲了 A ~ D 题, 课后必做作业是 F 题, 选做作业是 E 题)

## 课堂表现

今天课上讲的题目比较多, 同学们课上做题时间不是很多, 还有很多题后面空了很多题。

空题比较多的同学, 课下要好好做一做。

## 课堂内容

### P1387 最大正方形

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

using namespace std;

const int maxn = 100 + 5;
int p[maxn][maxn];

int get_sum(int x1, int y1, int x2, int y2) {
    return p[x2][y2] - p[x1-1][y2] - p[x2][y1-1] + p[x1-1][y1-1];
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            int x; cin >> x; p[i][j] = p[i-1][j] + p[i][j-1] - p[i-1][j-1] + x;
        }
    }

    int maxx = 0;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            for (int k = 1; k <= min(i, j); ++k) {
                if (get_sum(i-k+1, j-k+1, i, j) == k*k) maxx = max(maxx, k);
            }
        }
    }
    cout << maxx << endl;
    return 0;
}
```

### U477563 差为200的对数

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

using namespace std;

typedef long long LL;
int f[205];

int main()
{
    int n; cin >> n;
    LL res = 0;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x; f[x%200]++;
    }

    for (int i = 0; i <= 199; ++i) {
        res += ((LL)f[i]-1)*f[i]/2;
    }
    cout << res << endl;
    return 0;
}
```

### P3353 在你窗外闪耀的星星

先用桶数组记录每个位置上的值, 然后维护一个前缀和数组, 枚举找长度为  $w$  的最大区间和即可。

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

using namespace std;

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

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

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        int x, t; cin >> x >> t; w[x] += t;
    }
    for (int i = 1; i < maxn; ++i) p[i] = p[i-1] + w[i];

    int maxx = 0;
    for (int i = 1; i <= n; ++i) maxx = max(maxx, get_sum(i, i+m-1));
    cout << maxx << endl;
    return 0;
}
```

**P8649 [蓝桥杯 2017 省 B] k 倍区间**

当区间和是  $k$  的倍数, 说明  $(p[j] - p[i-1]) \% k == 0$

说明要找有多少  $p[j]\%k == p[i-1]\%k$  的方案

此时, 问题转成了 A 题, 可以按照 A 题的方法来做

需要额外注意的一点是, 这个题要用到  $p[0]$  点, 所以在维护桶数组是要把  $p[0]$  考虑上

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

using namespace std;

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

int main()
{
    int n, k; cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> w[i], p[i] = p[i-1] + w[i];

    for (int i = 0; i <= n; ++i) f[p[i]%k]++;

    LL res = 0;
    for (int i = 0; i < k; ++i) res += (LL)f[i]*(f[i]-1)/2;
    cout << res << endl;
    return 0;
}
```

**P6180 [USACO15DEC] Breed Counting S**

用 3 个一维数组, 或者用一个 3 行的 二维数组, 维护 1、2、3 每个数在每个位置是否出现

然后, 在维护前缀和数组, 就可以快速求一段区间内 1, 2, 3 的数量

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

using namespace std;

const int maxn = 1e5 + 5;
int w[maxn], p[4][maxn];

int get_sum(int id, int l, int r) { return p[id][r] - p[id][l-1]; }

int main()
{
    // ... (rest of the code)
}
```

```

int n, m; cin >> n >> m;
for (int i = 1; i <= n; ++i) cin >> w[i];

for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= 3; ++j) p[j][i] = p[j][i-1];
    p[w[i]][i]++;
}

while (m -- ) {
    int l, r; cin >> l >> r;
    for (int i = 1; i <= 3; ++i) cout << get_sum(i, l, r) << " ";
    cout << endl;
}
return 0;
}

```

### P4440 [COCI 2017/2018 #3] Programiranje

跟 D 题基本一致, 不过要维护 26 个一维数组, 或者是一个 26 行的 二维数组

```

#include <bits/stdc++.h>

using namespace std;

const int N = 50000 + 5, M = 26 + 5;
char s[N];
int p[M][N];

int get_int(char x) { return x-'a'+1; }

int get_sum(int id, int l, int r) { return p[id][r] - p[id][l-1]; }

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);
    for (int i = 1; i <= 26; ++i) {
        for (int j = 1; j <= n; ++j) {
            p[i][j] = p[i][j-1];
            if (get_int(s[j]) == i) p[i][j]++;
        }
    }

    int m; cin >> m;
    while (m -- ) {
        int l1, r1, l2, r2; cin >> l1 >> r1 >> l2 >> r2;
        bool flag = true;
        for (int i = 1; i <= 26; ++i) {
            if (get_sum(i,l1,r1) != get_sum(i,l2,r2)) flag = false;
        }
    }
}

```

```
    if (flag) cout << "DA" << endl;
    else cout << "NE" << endl;
}
return 0;
}
```