

前缀和、差分复习 + 二维数组模拟混练

人员

辛帅辰、高健桓、方俊喆、刘祺、夏硕承、秦显森、赵熙羽、牛同泽、齐振玮、徐浩然、谢亚锴、韩承睿 到课

作业检查

作业链接: <https://www.luogu.com.cn/contest/194988>

牟茗、赵熙羽、方俊喆、谢亚锴、齐振玮、辛帅辰 完成 4 道题

刘祺、夏硕承、牛同泽 完成 3 道题

高健桓 完成 2 道题

秦显森 完成 1 道题

作业

<https://www.luogu.com.cn/contest/197411>

课后作业为 D、E、F、G 4 道题

可能有个别题目比较好，同学们可以努力尝试一下，做不到满分也不要灰心

课堂表现

同学们上课积极性都比较高，听讲也很认真，提出表扬！！希望继续保持。

课堂内容

一维前缀和

```
int sum(int l, int r) {  
    return p[r] - p[l-1];  
}  
  
for (int i = 1; i <= n; i++) {  
    p[i] = p[i-1] + a[i];  
}
```

二维前缀和

```
int 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];  
}
```

```

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

```

一维差分

```

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

l~r: +k
p[l] += k, p[r+1] -= k;

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

```

U476049 数组填充

```

cin >> n
m = 2*n-1
x = m, y = n
for (int i = 1; i <= m*m; i++) {
    a[x][y] = i;

    1. 在最左下角 -> x--
    2. 在最后一行 -> x=1, y--
    3. 在最左边 -> y=m, x++
    4. 左下角有数字 -> x--
    5. 正常: x++, y--
}

```

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2000 + 5;
int a[maxn][maxn];

int main()
{
    int n; cin >> n;
    int m = 2*n-1;
}

```

```
int x = m, y = n;
for (int i = 1; i <= m*m; i++) {
    a[x][y] = i;
    if (x==m && y==1) {
        x--;
    }
    else if (x == m) {
        x = 1, y--;
    }
    else if (y == 1) {
        y = m, x++;
    }
    else if (a[x+1][y-1] != 0) {
        x--;
    }
    else {
        x++, y--;
    }
}

for (int i = 1; i <= m; i++) {
    for (int j = 1; j <= m; j++) {
        cout << a[i][j] << " ";
    }
    cout << endl;
}
return 0;
}
```

P8649 [蓝桥杯 2017 省 B] k 倍区间

维护一个前缀和 p 数组

当处理到第 i 个位置时, 若 $p[i] \% k == t$, 则要找前面有几个前缀和为 t 的

因为当 $p[j]\%k==t$, $p[i]\%k==t$ 时, $j+1 \sim i$ 这一段区间和就是 k 的倍数

所以可以用一个桶维护前面有几个位置的前缀和 取余k为t, 把这个数量存到 a[t] 中

所以在这个地方 答案 应该 $+= a[t]$

最后, 因为 t 多出现了一个, 所以让 $a[t]++$

```
// 核心代码
for (int i = 1; i <= n; ++i) {
    int t = p[i] % k;
    res += a[t];
    a[t]++;
}
```