# 错题集 1

## 1□Fake Maxpooling

链接

### 题意

给定一个 $n\times m$ 矩阵□$a_{i,j}=\text{lcm}(i,j)$□设 $f(i,j)=\{\max a_{x,y}|i\le x\lt i+k,j\le y\lt j+k\}$□求

\begin{equation}\sum_{i=1}^{n-k+1}\sum_{j=1}^{m-k+1}f(i,j)\end{equation}

### 题解

记忆化搜索求 $\text{gcd}$□两次单调队列维护最大值。

第一次单调队列维护每行长度为 $k$ 的区间的最大值，第二次单调队利用 $k$ 列每行长度为 $k$ 的区间的最大值。

时间复杂度 $O(nm)$□

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <algorithm>
#include <string>
#include <sstream>
#include <cstring>
#include <cctype>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <stack>
#include <queue>
#include <ctime>
#include <cassert>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset(a,b,sizeof(a))
using namespace std;
typedef long long LL;
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
```

```
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline char get_char(){
    char c=getchar();
    while(c==' '||c=='\n'||c=='\r')c=getchar();
    return c;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
const int MAXN=5005;
int a[MAXN][MAXN],b[MAXN][MAXN],que[MAXN];
int main()
{
    int n=read_int(),m=read_int(),k=read_int(),u,v;
    _rep(i,1,n)
    _rep(j,1,m){
        if(!a[i][j]){
            for(int k=1;k*i<=n&&k*j<=m;k++)
            a[i*k][j*k]=i*j*k;
        }
    }
    _rep(i,1,n){
        int front=1,tail=0;
        _for(j,1,k){
            while(front<=tail&&a[i][que[front]]<=a[i][j])front++;
            que[++tail]=j;
        }
        _rep(j,k,m){
            while(front<=tail&&j-que[front]>=k)front++;
            while(front<=tail&&a[i][que[front]]<=a[i][j])front++;
            que[++tail]=j;
            b[i][j]=a[i][que[front]];
        }
    }
    LL sum=0;
    _rep(j,k,m){
```

```
        int front=1,tail=0;
        _for(i,1,k){
            while(front<=tail&&b[que[front]][j]<=b[i][j])front++;
            que[++tail]=i;
        }
        _rep(i,k,n){
            while(front<=tail&&i-que[front]>=k)front++;
            while(front<=tail&&b[que[front]][j]<=b[i][j])front++;
            que[++tail]=i;
            sum+=b[que[front]][j];
        }
    }
    enter(sum);
    return 0;
}
```