2019icpc银川 K-Largest Common Submatrix

给两个$n\times m$的矩阵A,B，里面的数为$n*m$的排列，求最大相同子矩阵。

```
#include<bits/stdc++.h>
#define int long long
#define mk make_pair
#define pii pair<int,int>
#define F first
#define S second
using namespace std;
const int N=2007;

int n,m,A[N][N],B[N][N],mx[N][N],L[N][N],R[N][N];
pii pos[N*N];
int stk[N],top=0;

bool check(int x1,int y1,int x2,int y2){
    if(pos[B[x1][y1]].F-x1==pos[B[x2][y2]].F-x2&&pos[B[x1][y1]].S-y1==pos[B[x2]
[y2]].S-y2) return true;
    return false;
}

signed main(){
    ios::sync_with_stdio(0);
    cin.tie(0);cout.tie(0);
    cin>>n>>m;
    for(int i=1;i<=n;++i){
        for(int j=1;j<=m;++j){
            cin>>A[i][j];
            pos[A[i][j]].F=i;
            pos[A[i][j]].S=j;
        }
    }
    for(int i=1;i<=n;++i){
        for(int j=1;j<=m;++j){
            cin>>B[i][j];
        }
    }
    for(int i=1;i<=n;++i){
        for(int j=1;j<=m;++j){
            mx[i][j]=0;
            L[i][j]=R[i][j]=i; //能够向上和向下延申多少
            int a=pos[B[i][j]].F,b=pos[B[i][j]].S;
            while(b+mx[i][j]<=m&&j+mx[i][j]<=m&&A[a][b+mx[i][j]]==B[i][j+mx[i]
[j]]) ++mx[i][j];
        }
    }
    for(int j=1;j<=m;++j){
        for(int i=2;i<=n;++i){
            if(mx[i-1][j]>=mx[i][j]&&check(i,j,i-1,j)) L[i][j]=L[i-1][j];
            while(L[i][j]-1>=1&&mx[i][j]<=mx[L[i][j]-1][j]){
                if(!check(i,j,L[i][j]-1,j)) break;
```

```
                --L[i][j];
            }
        }
        for(int i=n-1;i>=1;--i){
            if(mx[i+1][j]>=mx[i][j]&&check(i,j,i+1,j)) R[i][j]=R[i+1][j];
            while(R[i][j]+1<=n&&mx[i][j]<=mx[R[i][j]+1][j]){
                if(!check(i,j,R[i][j]+1,j)) break;
                ++R[i][j];
            }
        }
    }
    int ans=0;
    for(int i=1;i<=n;++i){
        for(int j=1;j<=m;++j){
            ans=max(ans,mx[i][j]*(R[i][j]-L[i][j]+1));
        }
    }
    cout<<ans<<"\n";
    return 0;
}
```