

# COGS 4160 hope I can sort matrix

大意就是判断通过两种操作判断是否能变为不减序列，特殊的 0 视为万能字符。

0 的问题可以先不考虑。

一个信息：行列是独立的，即每行的数本质是**不变的**（最大值最小值不变）。

所以第一个条件就是需要满足行行之间的不减关系。

这样行就固定了，只需要解决每行内不减关系，只有一行是好解决的，我们需要考虑多行的相互关系。

本质上每行给定了一些列的偏序关系，我们考虑通过根据这些偏序关系进行排序（顺带判断两列是否能够同时满足各自的关系），简单一点的，可以利用 sort 的 cmp 函数进行比较。

这样只要排序后满足所有的关系，即可找到第一组解，否则无解。

考虑上 0，万能字符一定要万能，说明存在 0 的地方偏序关系可以是任意关系，特判即可。

复杂度  $\mathcal{O}(nm \log nm)$ 。

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define ull unsigned long long
#define db double
#define pii pair<int,int>
#define pll pair<ll,ll>
#define F(i,l,r) for(int i = (l);i <= (r);++i)
#define D(i,r,l) for(int i = (r);i >= (l);--i)
#define vi vector<int>
#define mem(a,x) memset(a,x,sizeof a)
#define fi first
#define in inline
#define se second
#define pb push_back
#define bug cerr<<"LINE:"<<__LINE__<<" "
mt19937 rd(std::chrono::steady_clock::now().time_since_epoch().count());
constexpr int N = 1e6+10;
constexpr ll inf = 1e18,mod = 998244353;

ll read(){
    ll x = 0,f = 1;char c = getchar();
    for(;c < '0' || c > '9';c = getchar())if(c == '-')f = -1;
    for(;c >= '0' && c <= '9';c = getchar())x = (x<<1) + (x<<3) + c-'0';
    return x * f;
}

void cmax(ll &x,ll y){x = max(x,y);}
void cmin(ll &x,ll y){x = min(x,y);}

int n,m;
int id[N];
```

```

bool flag = 1;
vi a[N];
vector<pii> s;
bool cmp(int x,int y){
    int now = -1;
    for(int i = 1;i <= n;i++){
        if(!a[i][x] || !a[i][y] || a[i][x] == a[i][y])continue;
        int t = a[i][x] < a[i][y];
        if(now == -1)now = t;
        if(t != now)return flag = 0,x < y;
    }
    if(now == -1)return x < y;
    return now;
}
void solve(){
    n = read(),m = read();
    F(i,1,n){
        a[i].resize(m + 1);
        int mi = 1e8,mx = 0;
        F(j,1,m){
            a[i][j] = read();
            if(a[i][j])mi = min(mi,a[i][j]),mx = max(mx,a[i][j]);
        }
        if(mx != 0)s.pb({mi,mx});
    }
    sort(s.begin(),s.end());int la = 0;
    for(auto x : s){
        if(la > x.fi)return printf("No\n"),void();
        la = x.se;
    }
    F(i,1,m)id[i] = i;
    sort(id + 1,id + 1 + m,cmp);
    F(i,1,m - 1)cmp(i,i + 1);
    puts(flag ? "Yes" : "No");
}

int main(){
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    int t = 1;
    while(t--)solve();
    return 0;
}

```

