

暴力遍历每一种列和行组合，对于每一种组合，先判断其是否符合条件，对于满足条件的不难发现最小操作数就是逆序对数量，但是此时求逆序对数量可以 $O(N^2)$ 解决，所以可以使用类似冒泡排序的思想快速求出，详细细节可以看代码。

```
void solve(){
    int n,m;
    cin >> n >> m;
    vector<vector<LL>> a(n + 1,vector<LL>(m + 1)),b(n + 1,vector<LL>(m + 1));
    for(int i = 1;i <= n;i ++){
        for(int j = 1;j <= m;j ++){
            cin >> a[i][j];
        }
    }
    for(int i = 1;i <= n;i ++){
        for(int j = 1;j <= m;j ++){
            cin >> b[i][j];
        }
    }
    vector<int> raw(n + 1),col(m + 1);
    for(int i = 1;i <= n;i ++){
        raw[i] = i;
    }
    for(int i = 1;i <= m;i ++){
        col[i] = i;
    }
    auto check = [&](vector<int> r,vector<int> c) -> int{
        for(int i = 1;i <= n;i ++){
            for(int j = 1;j <= m;j ++){
                if(a[r[i]][c[j]] != b[i][j]){
                    return INF;
                }
            }
        }
    };
    int num = 0;
    for(int i = 1;i <= n;i ++){
        for(int j = 2;j <= n - i + 1;j ++){
            if(r[j - 1] > r[j]){
                swap(r[j],r[j - 1]);
                num ++;
            }
        }
    }
    for(int i = 1;i <= m;i ++){
        for(int j = 2;j <= m - i + 1;j ++){
            if(c[j - 1] > c[j]){
                swap(c[j],c[j - 1]);
                num ++;
            }
        }
    }
}
```

```

    }
}
return num;
};
int ans = INF;
do{
    do{
        ans = min(ans, check(raw, col));
    }while(next_permutation(col.begin() + 1, col.end()));
}while(next_permutation(raw.begin() + 1, raw.end()));
if(ans == INF) cout << -1 << endl;
else cout << ans << endl;
}

```