

Codeforces Round #704 (Div. 2)

[比赛链接](#)

E. Almost Fault-Tolerant Database

题意

给定 n 个长度为 m 的数组 $\{a_i\}$ 要求找到一个长度为 m 的数组 b 使得 b 与每个 $\{a_i\}$ 不同的位置数不超过 2 。

题解

首先如果所有 $\{a_i\}$ 与 $\{a_1\}$ 不同的位置数都不超过 2 ，则 $\{a_1\}$ 就是答案。

否则如果存在 $\{a_j\}$ 与 $\{a_1\}$ 不同的位置数大于 4 ，则无解。

如果存在 $\{a_j\}$ 与 $\{a_1\}$ 不同的位置数等于 4 ，则暴力枚举两个一定要修改的位置，修改后暴力验证答案，如果全部失败则无解。

如果存在 $\{a_j\}$ 与 $\{a_1\}$ 不同的位置数等于 3 ，则暴力枚举一个一定要修改的位置和一个可能要修改的位置。

先修改一定要修改的位置，然后验证答案，如果所有 $\{a_i\}$ 与当前答案不同的位置数都不超过 2 ，则成功。

否则发现一个数组 $\{a_k\}$ 与当前答案不同位置大于 2 ，则将可能要修改的位置修改为 $\{a_k\}$ 位置对应的值，此时答案固定，重新验证。

如果暴力枚举一定要修改的位置和可能要修改的位置全部失败则无解。

总时间复杂度 $O(nm)$

```
const int MAXN=2.5e5+5;
vector<int> a[MAXN];
vector<int> cmp(vector<int> a,vector<int> b){
    vector<int> diff;
    _for(i,0,a.size()){
        if(a[i]!=b[i])
            diff.push_back(i);
    }
    return diff;
}
void check(vector<int> ans,int n){
    _for(i,1,n){
        vector<int> diff=cmp(ans,a[i]);
        if(diff.size()>2)return;
    }
}
```

```
}
puts("Yes");
_for(i,0,ans.size())space(ans[i]);
exit(0);
}
int main()
{
    int n=read_int(),m=read_int();
    _for(i,0,n){
        a[i].resize(m);
        _for(j,0,m)
            a[i][j]=read_int();
    }
    check(a[0],n);
    _for(i,1,n){
        vector<int> diff=cmp(a[0],a[i]);
        if(diff.size()>=3){
            if(diff.size()==4){
                vector<int> ans=a[0];
                _for(j,0,4)_for(k,j+1,4){
                    ans[diff[j]]=a[i][diff[j]];
                    ans[diff[k]]=a[i][diff[k]];
                    check(ans,n);
                    ans[diff[j]]=a[0][diff[j]];
                    ans[diff[k]]=a[0][diff[k]];
                }
            }
            else if(diff.size()==3){
                vector<int> ans=a[0];
                _for(j,0,3)_for(k,0,3){
                    if(j==k)continue;
                    ans[diff[j]]=a[i][diff[j]];
                    check(ans,n);
                    _for(t,1,n){
                        vector<int> diff2=cmp(ans,a[t]);
                        if(diff2.size()>=3){
                            if(diff2.size()==3&&find(diff2.begin(),diff2.end(),diff[k])!=diff2.end()){
                                ans[diff[k]]=a[t][diff[k]];
                                check(ans,n);
                                ans[diff[k]]=a[0][diff[k]];
                            }
                            break;
                        }
                    }
                    ans[diff[j]]=a[0][diff[j]];
                }
            }
            break;
        }
    }
    break;
}
}
```

```
puts("No");  
return 0;  
}
```

From:

<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_704_div._2 

Last update: **2021/02/24 16:17**