

安排公牛

一. 考察内容:

动态规划 状态压缩

二. 题目分析:

[题目大意]

给出 n ($n \leq 20$) 头牛, 和他们喜欢的房间, 要求求出有多少种分配数, 使得每头牛都独自待在自己喜欢的房间内。

[写题思路]

考虑用动态规划实现, 设 $f[i][j]$ 为前 i 头牛, 住在 j 集合的房间中的方案数, 很显然, 第 i 头牛可以选择一个自己喜欢的房间入住, 所以 $f[i][j] = f[i-1][j - \text{love}(i)]$ 。初始状态为 $f[0][0] = 0$ 。

由于本题空间限制比较小, 所以考虑将第一位压缩掉, 然后将第二位倒序枚举。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 21
*文件类型: 源代码文件
*题目来源: COGS
*当前状态: 已通过
*备忘录: 动态规划 状态压缩
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys[表情]pes.h>
// #include <unistd.h>
using namespace std;
#define MAXN 21
int n, m;
bitset <MAXN> love[MAXN];
unsigned int f[1<<MAXN];
/* Variable explain:
f[i]: i集合的牛的方案数
*/
void read()
{
    int ls1, ls2;
    scanf("%d%d", &n, &m);
    for(int i=1; i<=n; ++i)
    {
        scanf("%d", &ls1);
        for(int j=1; j<=ls1; ++j) scanf("%d", &ls2), love[i][ls2]=1;
    }
    return;
}
int num(int a)
{
    int ans=0;
    while(a){if(a&1)++ans; a>>=1;}
    return ans;
}
```

```
bool in(int a,int b)
{
    return (a&(1<<b));
}
void dp()
{
    int ans=0;
    //f[i][j]为前i个牛, 安排的j集合的牛棚时的方案数, 将第一维压缩掉
    f[0]=1;
    int U=(1<<m)-1;
    for(int i=1;i<=n;++i)
    {
        for(int j=U;j;--j)
        {
            if(num(j)!=i)continue;
            for(int k=0;k<m;++k)
            {
                if(!in(j,k))continue;
                if(love[i][k+1]) f[j]+=f[j-(1<<k)];
            }
            if(i==n)ans+=f[j];
            printf("f[%d][%d]=%d\n",i,j,f[j]);
        }
    }
    printf("%d\n",ans);
}

int main()
{
    // freopen("examnine.in","r",stdin);
    // freopen("examnine.out","w",stdout);
    read();
    dp();
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)