Atcoder Rugular Contest 106

比赛链接

E - Medals

题意

给定 n 个员工，每个员工第 $[2k*a_i+1,(2k+1)*a_i]$ 天上班，第 $[(2k+1)*a_i+1,(2k+2)*a_i]$ 天休息。

每天最多可以给一名当天上班的员工一个奖章，问最少需要多少天才能使每个员工至少有 k 个奖章。

题解

建立二分图，左部 $n*k$ 个点代表每个员工的每个奖章，右部为天数，然后每个员工的奖章向该员工对应的上班时间连边。

于是问题转化为二分图匹配问题，考虑二分天数，然后检查左部是否存在完全匹配。

接下来给出二分图完全匹配的 Hall 定理：

设左部集合为 U，$T(S)$ 表示右部中所有与 S 有连边的点构成的集合。则左部可以完全匹配等价于对于任意非空集合 $S\subset U$，有 $|S|\le |T(S)|$。

回到原题，发现 $T(S)$ 的大小至于 S 中包含的员工的种类数有关，与每种员工有多少个奖章无关。

于是不妨只考虑每个员工的奖章要么不选，要么全选的情况，因为这样可以保证 $T(S)$ 不变时 $|S|$ 最大。

发现 $T(S)$ 难以直接求解，考虑求 $T(S)$ 的补集，这等价于总天数 $-$ 仅含 $U-S$ 的子集(包括空集，即无人上班)的员工上班的天数。

先求出每天代表的上班员工的集合，然后用桶维护每个员工集合的上班天数，最后维护一下子集和即可，子集和的维护具体见代码。

最后，关于二分的上下界，首先不难发现下界为 nk，另外对于 $2nk$ 天每个员工至少上班 nk 天，于是 $|T(S)|\ge |S|$，所以 $2nk$ 为上界。

总时间复杂度 $O((n2^n+nk)\log nk)$。

```
const int MAXN=20,MAXS=1<<MAXN,MAXC=2e5*MAXN;
int a[MAXN],c[MAXS],minc[MAXS],d[MAXC];
int main()
{
    int n=read_int(),k=read_int(),U=(1<<n)-1;
    _for(i,0,n)a[i]=read_int();
```

```
    _for(i,0,MAXC){
        _for(j,0,n){
            if((i/a[j])%2==0)
            d[i]|=1<<j;
        }
    }
    _rep(i,1,U)minc[i]=minc[i&(i-1)]+k;
    int lef=n*k,rig=2*n*k,ans;
    while(lef<=rig){
        int mid=lef+rig>>1;
        mem(c,0);
        _for(i,0,mid)c[d[i]]++;
        _for(i,0,n){
            _for(j,0,1<<n){
                if(j&(1<<i))
                c[j]+=c[j^(1<<i)];
            }
        }
        bool flag=true;
        _rep(i,1,U){
            if(minc[i]>mid-c[U^i]){
                flag=false;
                break;
            }
        }
        if(flag){
            ans=mid;
            rig=mid-1;
        }
        else
        lef=mid+1;
    }
    enter(ans);
    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:arc_106&rev=1613363902

Last update: **2021/02/15 12:38**