

Atcoder Rugular Contest 106

[比赛链接](#)

E - Medals

题意

给定 n 个员工，每个员工第 $[2k \cdot a_i + 1, (2k+1) \cdot a_i]$ 天上班，第 $[(2k+1) \cdot a_i + 1, (2k+2) \cdot a_i]$ 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 $n \cdot k$ 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 $S \subseteq U$ 有 $|S| \leq |T(S)|$

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 $|T(S)| \geq |S|$ 所以 $2nk$ 为上界。

总时间复杂度 $O((n^2 + nk) \log nk)$

```
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;
}
```

F - Figures

题意

给定 n 个带标号点，每个点有 a_i 个不同的接口，每个接口最多有一个度，问生成树的个数。

题解

设每个点度数为 d_i 于是在每个生成树中每个点的贡献为 $a_i^{\underline{d_i}}$ 根据 prufer 序列性质，答案为

$$\sum_{\sum d_i=2n-2} \binom{n-2}{d_1-1, d_2-1, \dots, d_{n-1}} \prod_{i=1}^n a_i^{\underline{d_i}} = \sum_{\sum d_i=2n-2} \frac{(n-2)!}{(d_1-1)!(d_2-1)! \dots (d_{n-1})!} \prod_{i=1}^n a_i^{\underline{d_i}}$$

发现这东西可以用指数型生成函数搞，为方便处理，令 $c_i = d_i + 1$ 于是上式化为

$$\sum_{\sum c_i=n-2} \binom{n-2}{c_1, c_2, \dots, c_n} \prod_{i=1}^n a_i^{\underline{c_i+1}}$$

考虑 $a_i = k$ 的点对应的生成函数，枚举 $c_i \in [0, k-1]$ 贡献为 $k^{\underline{c_i+1}}$ 于是有

$$\begin{aligned} f_k(x) &= \sum_{i=0}^{k-1} \frac{k^{\underline{i+1}}}{i!} x^i \\ &= \sum_{i=0}^{k-1} \frac{k!}{i!(k-1-i)!} x^i \\ &= k(x+1)^{k-1} \end{aligned}$$

答案为
$$\begin{aligned} & \sum_{\sum d_i=2n-2} \frac{(n-2)!}{d_1! d_2! \dots d_{n-1}!} \prod_{i=1}^n a_i^{\underline{d_i+1}} \\ &= (n-2)! \prod_{i=1}^n a_i^{\underline{d_i+1}} (x+1)^{\sum (a_i-1)} \\ &= (n-2)! \prod_{i=1}^n a_i^{\underline{d_i+1}} \left(\sum_{i=1}^n a_i \right)^{n-2} \end{aligned}$$

时间复杂度 $O(n)$

```
const int Mod=998244353;
int main()
{
    int n=read_int(),s=Mod-n,ans=1;
    _for(i,0,n){
        int a=read_int();
        ans=1LL*ans*a%Mod;
        s=(s+a)%Mod;
    }
    _for(i,0,n-2)
        ans=1LL*ans*(s+Mod-i)%Mod;
    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

Last update: 2021/02/15 17:06