

codeforces round 647

A Johnny and Ancient Computer

题意：给一个数 x 可以对它进行6种操作，乘2，乘4，乘8，除2，除4，除8，问至少要多少次操作才能把 x 变成 a ？

题解：如果能通过以上6种操作互相得到，那么二进制的表达从开头到末尾必定只有0的个数不同，然后算出0的个数差，除3（每次消去或增加3位一定是最优的）向上取整即可。

B Johnny and His Hobbies

题意：给 n 个数，问最小的数 k ，使得，这 n 个数中每个数分别与 k 取异或后得到的数组成的集合满足和原来的集合完全相同。

题解：妙就妙在数据范围上，小于1000的数据范围直接枚举+暴力即可。

C Johnny and Another Rating Drop

题意：给一个数 n 从0开始，每次对数加1，假设加完后数为 x 将 x 和 $x-1$ 化为2进制，并从低位开始比较，不足补0，遇到不同的位，则将答案加1，问加到 n 时，这个答案能有多少，对 10^9+7 取模。

题解：找规律，列举几十个后会发现，当 n 为2的 i 次幂时，答案便是 2^i-1 不难发现，一个数 n 可以写成若干2的次幂的和，则大难即为对应的2的次幂情况的和（因为从低位加到高位时，高位的数字不会变化，理解不了可以直接写几个找规律），然后便可以 $O(n \log n)$ 求得答案。

D Johnny and Contribution

题意：有一张图，要对其进行染色，染色的规则如下：从一个点开始，每次先观察与概念相连的已经染过色的所有点的值，然后取**总体的**最小值，每个点有个目标值，问最后能否存在一个方法，让每个点得到目标的染色值？

题解：这意思真的绕的不行……，绕出来之后发现还可以，首先，先不管目标第一个涂得点一定只能是1，而且有1才有2，这样不难想到，先将所有点按照目标值从小到大排序，先将所有点更新为1，然后遍历点，每次将点周围的点满足和该点（目前的值）相同的点的颜色更新（即+1），但凡遍历到一个点满足目标值不等于现有值，一定是不满足条件的。

E Johnny and Grandmaster

题意：给一个数 p ，还给了 n 个数 $\{k_1, k_2, \dots, k_n\}$ ，现在将 $\{p^{k_1}, p^{k_2}, \dots, p^{k_n}\}$ 分成两组，要使得两组和的差的绝对值最小，问最小为多少，最小对 10^9+7 取模？

题解：原题显然可以转化为选择一些数取正，将另一些数取负，之后加在一起，要和的绝对值最小。首先

有个引理：对于上述 p^k 次幂的集合，将其从大到小排序，若满足 $p^{k_1} < p^{k_2} + p^{k_3} + \dots + p^{k_n}$ ，则必定存在 m 使得 $p^{k_1} = p^{k_2} + p^{k_3} + \dots + p^{k_m}$ ，这是显然的，讲所有数写成 p 进制，发现这些数最高位全是1，若想从小的加成的大的，必须一位一位的进位，中间总有一种状态为 p^{k_1} ，这样就很简单了，先将 k 数组从大到小排序，开一个数 now 存目前的值，每当 now 为0就加，否则则减，这样得到的值一定是最小的，注意处理过程中开两个模数去摸，避免倍数差被模数抵消。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int maxn=1e6+13;
int mod1=1e9+7,mod2=1e9+3;//这个为了防止去摸的时候产生倍数差
int k[maxn];
ll qpow(int a,int b,int mod)
{
    int ans=1;
    int p=a;
    while(b)
    {
        if(b&1)
        {
            ans=1ll*ans*p%mod;
        }
        p=1ll*p*p%mod;
        b>>=1;
    }
    return 1ll*ans;
}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        ll ans1=0,ans2=0;
        int n,p;
        scanf("%d%d",&n,&p);
        for(int i=1;i<=n;i++)
        {
            scanf("%d",&k[i]);
        }
        sort(k+1,k+1+n);
        for(int i=n;i;i--)
        {
            if(!ans1&&!ans2)
            {
                ans1+=qpow(p,k[i],mod1);
                ans2+=qpow(p,k[i],mod2);
            }
        }
    }
}
```

```
        else
        {
            ans1-=qpow(p,k[i],mod1);
            ans1=(ans1%mod1+mod1)%mod1;
            ans2-=qpow(p,k[i],mod2);
            ans2=(ans2%mod2+mod2)%mod2;
        }
    }
    printf("%lld\n",ans1);
}
```

From:

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

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforces_round_647_div2 

Last update: **2020/06/27 17:58**