

I题解题报告：并查集(路径压缩优化)

题意：多次合并集合、查询两点是否再同一集合内

很明显是用并查集解决问题。

本题难点在于如何最高效地使用并查集。

我们知道，并查集可以压缩路径。

如果我们不压缩路径，并查集可能会被卡成一条链，导致并查集退化。

如果单纯地压缩路径，也很容易被卡。

这里我们利用启发式合并的思想，记录每个集合的深度，合并的时候就把深度小的集合并到深度大的集合上。

这样可以很大限度的减少集合深度扩张的速度。

具体代码见下：

```
void joint(int x,int y){
    int prex=find(x),prey=find(y);
    if(prex==prey) return;
    if(s[prex]<=s[prey]){
        pre[prex]=prey;
        ++s[prey]; //s[]里存的就是深度
        return;
    }
    pre[prey]=prex;
    ++s[prex];
}
```

另外注意到这个题目是个很长的二进制数取模，

我们可以把二进制数拆成很多个数相加，

比如说十进制12453可以拆成10000+2000+400+50+3，

通过打表的方法把10，100，1000，10000……这些二进制取模后的值存起来，

最后算总和的时候依据具体情况进行加，取模操作。

```
void build(int lim){//二进制取模打表
    two_mod[lim]=1;
    for(int i=lim-1;i>0;--i){
        two_mod[i]=((two_mod[i+1])<<1)%mod;
    }
}
```

时间复杂度 $O(n \log n)$:

并查集启发式合并树高最高为 $\log n$ ， n 次查询共 $n \cdot \log n$ 次。合并其实就是查询。