**题目大意**：初始时给n个集合，第i个集合有一个数ai。现在有三种操作：

1 u v：将第u个数所在的集合和第v个数所在的集合合并

2 u ：将第u个数所在集合里的所有数全部加1

3 u k x：询问第u个数所在的集合中有多少个数%2^k=x

对于每一个3操作输出对应的答案。

**题解**：对于3操作，实际上就是询问集合中二进制后缀为某值的数有多少个，于是每个集合从低位到高位建立trie树（和平时建立trie树相反），即可完成查询操作。对于1操作可以使用trie树合并做到nlogai的复杂度。对于2操作，每次+1时相当于将所有数最低位的0变成1,1变成0，再将所有最低位是1的次低位0变成1,1变成0，再将所有最低位和次低位是1的第三低位0变成1,1变成0……即在建出的trie树上0儿子和1儿子，再递归处理原来是1的儿子继续交换。因为始终只要递归一边，这样可以做到在log的时间里实现集合全部加一。这样既可在O(nlogai)的复杂度内解决本题。