有一个集合，在一次操作中你可以选择集合中的一个元素然后将它和x相与

问是否有可能在施加几次操作之后产生与集合中其他元素相等的数，如果可能，输出最小的操作数,如果不可能输出-1

操作数最多为2，再多操作是无用的，因为a&x再&x结果还是a

1. 次数为1的情况举例：

a,b,c,d,e,f

a&x---->d

1. 次数为2的情况举例：

a,b,c,d,e,f

a&x---->t

b&x---->t

不会有更多次数的操作使得产生与集合中其他元素相等的数了

但这个题在比赛的时候自己一直被卡住并且打死都找不到原因

错误代码：

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

set<int> \_set;

set<int>set2;

set<int>::iterator it;

map<int,int>\_map;

int main()

{

int n,x;

cin>>n>>x;

int t;

bool flag=0;

for(int i=0;i<n;i++)

{

cin>>t;

\_set.insert(t);

\_map[t]++;

if(\_map[t]>1)

flag=1;

}

if(flag)

{

cout<<0<<"\n";

return 0;

}

for(it=\_set.begin();it!=\_set.end();it++)

{

int e=(\*it)&x;

if(e==(\*it))continue;

if(\_set.find(e)!=\_set.end() && \_set.find(e)!=it)

{

cout<<1<<"\n";

return 0;

}

else

if(set2.find(e)!=set2.end())

{

cout<<2<<"\n";

return 0;

}

else

set2.insert(e);

}

cout<<-1<<"\n";

return 0;

}

想法是第一次扫的时候把与集合元素新产生的数加入到另一个集合2中，第二次再扫集合看集合元素与x是否在集合2中

但你忽略了一个问题

元素1&x产生的结果y

在第二次扫集合的时候元素1&x还是y，这样肯定就在集合2中了！！！

解题代码

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

set<int> \_set;

set<int>set2;

set<int>::iterator it;

int main()

{

//freopen("input.txt","r",stdin);

int n,x;

cin>>n>>x;

int t;

bool flag=0;

for(int i=0;i<n;i++)

{

cin>>t;

\_set.insert(t);

}

if(\_set.size()<n)

{

cout<<0<<"\n";

return 0;

}

int e;

for(it=\_set.begin();it!=\_set.end();it++)

{

e=(\*it)&x;

if(e==(\*it))continue;

if(\_set.find(e)!=\_set.end())

{

cout<<1<<"\n";

return 0;

}

}

for(it=\_set.begin();it!=\_set.end();it++)

{

e=(\*it)&x;

if(set2.find(e)!=set2.end())

{

cout<<2<<"\n";

return 0;

}

set2.insert(e);

}

cout<<-1<<"\n";

return 0;

}