构造函数

bitset<n> b;

b有n位，每位都为0.参数n可以为一个表达式.

如bitset<5> b0;则"b0"为"00000";

bitset<n> b(unsigned long u);

b有n位,并用u赋值;如果u超过n位,则顶端被截除

如:bitset<5>b0(5);则"b0"为"00101";

bitset<n> b(string s);

b是string对象s中含有的位串的副本

string bitval ( "10011" );

bitset<5> b0 ( bitval4 );

则"b0"为"10011";

bitset<n> b(s, pos, num);

b是s中从位置pos开始的num个位的副本,如果num<n,则前面的空位自动填充0;

string bitval ("11110011011");

bitset<6> b0 ( bitval5, 3, 6 );

则"b0" 为 "100110";

os << b

把b中的位集输出到os流

os >>b

输入到b中,如"cin>>b",如果输入的不是0或1的字符,只取该字符前面的二进制位.

bool any( )

是否存在置为1的二进制位？和none()相反

bool none( )

是否不存在置为1的二进制位,即全部为0？和any()相反.

size\_t count( )

二进制位为1的个数.

size\_t size( )

二进制位的个数

flip()

把所有二进制位逐位取反

flip(size\_t pos)

把在pos处的二进制位取反

bool operator[]( size\_type Pos )

获取在pos处的二进制位

set()

把所有二进制位都置为1

set(pos)

把在pos处的二进制位置为1

reset()

把所有二进制位都置为0

reset(pos)

把在pos处的二进制位置为0

注意：bitset只能与bitset运算，不能与数运算

|  |
| --- |
| /\*  题意：给出n个集合(n<=1000),每个集合中最多有10000个数，  每个数的范围为1~10000，给出q次询问(q<=200000),  每次给出两个数u，v判断是否有一个集合中同时含有u，v两个数  \*/  #include<iostream>  #include<bitset>  #include<stdio.h>  using namespace std;  bitset<1001>b[10010];  int main()  {  int n,c,q;  int p,x,y;  cin>>n;  for(int i=1;i<=n;i++)  {  scanf("%d",&c);  for(int j=1;j<=c;j++)  {  scanf("%d",&p);  b[p].set(i);  }  }  cin>>q;  for(int i=1;i<=q;i++)  {  scanf("%d%d",&x,&y);  int flag=0;  if((b[x]&b[y]).any())flag=1;  if(flag)cout<<"Yes"<<endl;  else cout<<"No"<<endl;  }  return 0;  } |