//Union Find Disjoint Set

//Data Structure

vector<int>u\_list, u\_set;

//u\_list[x] contains the size of a set x

//u\_set[x] contains the root of x

int unionRoot(int n) {

if(u\_set[n] == n)

return n;

else

return u\_set[n] = unionRoot(u\_set[n]);

}

int makeUnion(int a, int b) {

int x = unionRoot(a);

int y = unionRoot(b);

if(x == y)

return x;

else if(u\_list[x] > u\_list[y]) {

u\_set[y] = x;

u\_list[x] += u\_list[y];

return x;

}

else {

u\_set[x] = y;

u\_list[y] += u\_list[x];

return y;

}

}

void unionInit(int len) {

u\_list.resize(len+5, 1);

u\_set.resize(len+5);

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

u\_set[i] = i;

}

bool isSameSet(int a, int b)

{

if(unionRoot(a) == unionRoot(b))

return 1;

return 0;

}