Key definitions:

A ***binary relation*** R for a set S is a set of SxS or $R \subseteq SxS$

1) For example S=N (natural numbers) R is a divisor of 3 is a divisor of 6, 3R6
2) < for real numbers R
   3 is less than 6, 3<6

A relation is

- ***reflexive*** if aRa for all $a \in S$
- ***Symmetric*** if aRb <=> bRa
- ***Transitive*** if aRb and bRc => aRc

	Reflexive	Symmetric	Transitive
Is a divisor of	Yes	No	Yes
<	No	No	Yes
$\leq$	Yes	No	Yes

Assume all the undergraduate students have only one advisor, sharing the same advisor is a binary relation defined on the set of students

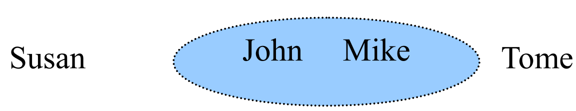
Sharing the same advisor
Reflexive? Yes
Symmetric? Yes
Transitive? Yes

We call a reflexive, symmetric, and transitive relation an ***equivalence*** relation. If we have an equivalence relation we could "partition" the set into a group of subsets such that

$S1 \cup S2 ..... \cup Sn = S$

$Si \cap Sj = \{ Si$ if i=j for all $(i,j) \in [1,n] \}$ and $\varnothing$ otherwise

for example
Susan → Dr. Wang
Mike → Dr. Smith
Tome → Dr. Singh
John → Dr. Smith



Disjoint set is a data structure for equivalence relation between two operations:

1) find returns a unique ID such that find(x)=find(y) if and only if x and y belong to the same class
2) Union merges two classes.

   For the time being lets assume that the elements are called 0,1,2,3,....k $k \in N$

Design options

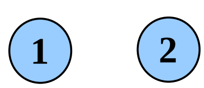
Hash table (or look up table)

0
1
2
3
4
5

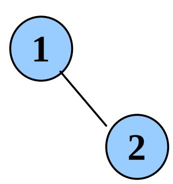
Find O(1)
Union O(n)

Tree representation

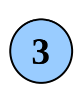
merge



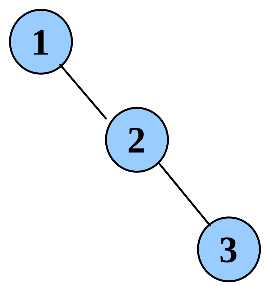
gives



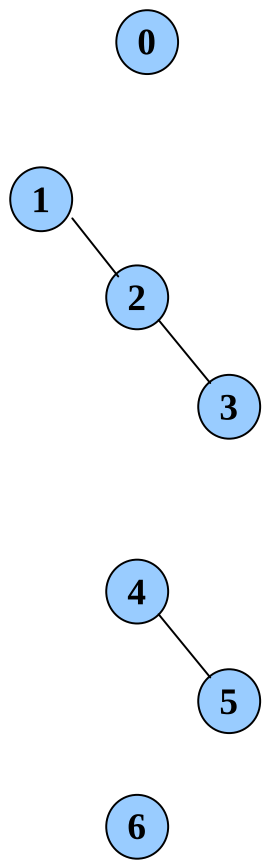
merge



Gives

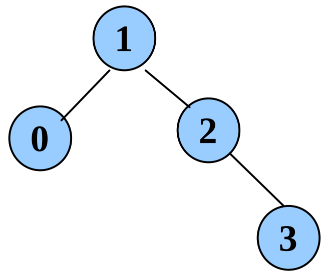


Find operation returns the root of the tree.

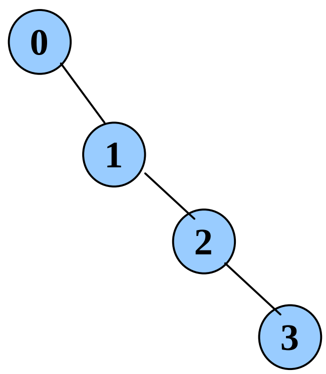


find (1)	1
(0)	0
(2)	1
(3)	1
(4)	4
(5)	4
(6)	6

union (0,3)



or



union (1,3)

Nothing