

Precondition on MakeSet(x):
 x does not already belong
to another set.

Union(x, y)

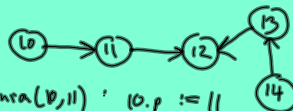
(representative: 1st element)
append list containing x onto
list containing y .

Find(x) — make use circularly
linked list.
(or doubly linked)

Union in constant time if
doubly linked, circular list,
or doubly linked list with
pointers to 1st & last items.
But must find those, so actually
linear time.

Each set is a tree of items
in the set. Only link in each
node is to the parent.

Root of tree is representative.
no parent ptr.



Union(10, 11) : 10.p := 11
11 is the rep.

Union(10, 12)

Union(13, 14)

Union(14, 11)

Find(x): follow parent pointers
from x until root.

This is better, but still not
good enough.

Two additional tricks:

1. path compression

During a Find:



Second trick: when merging



make tree with smaller depth
child of tree with larger depth.

Need to store the tree height
at each root.

d_1 x y d_2

if $d_1 < d_2$, then $x.p = y$

if $d_1 > d_2$, then $y.p = x$

if $d_1 = d_2$, then $x.p = y$
 $y.p = x$

Problem: ~~next~~ next time: