Disjoint Sets

Prem Nair

DisjointSets Data Structure

- Data structure for maintaining a partition of a set into disjoint subsets (data structure sometimes called *Partition* rather than DisjointSets)
- General features
  - *Data:*
    - Universe U – the base set that is being partitioned (this set is never altered)
    - Collection C = {X_1, X_2, …, X_n} of subsets of the universe – the subsets are disjoint and their union is U (these subsets are modified when the data structure is used – size of C shrinks because of repeated union operations)
  - *Operations:*
    - find(x) – returns the subset X_i to which x belongs
    - union(A,B) – replaces the subsets A, B in C with A U B.

Array based Implementation of DisjointSets as Trees.

- The elements of each set X in the collection C are represented by nodes in a tree T_X; the set X itself is referenced by its root r_X.
- find(x) returns the root of the tree to which x belongs
- union(x,y) joins the tree x belongs to to the tree y belongs to by pointing root of one to the root of the other.

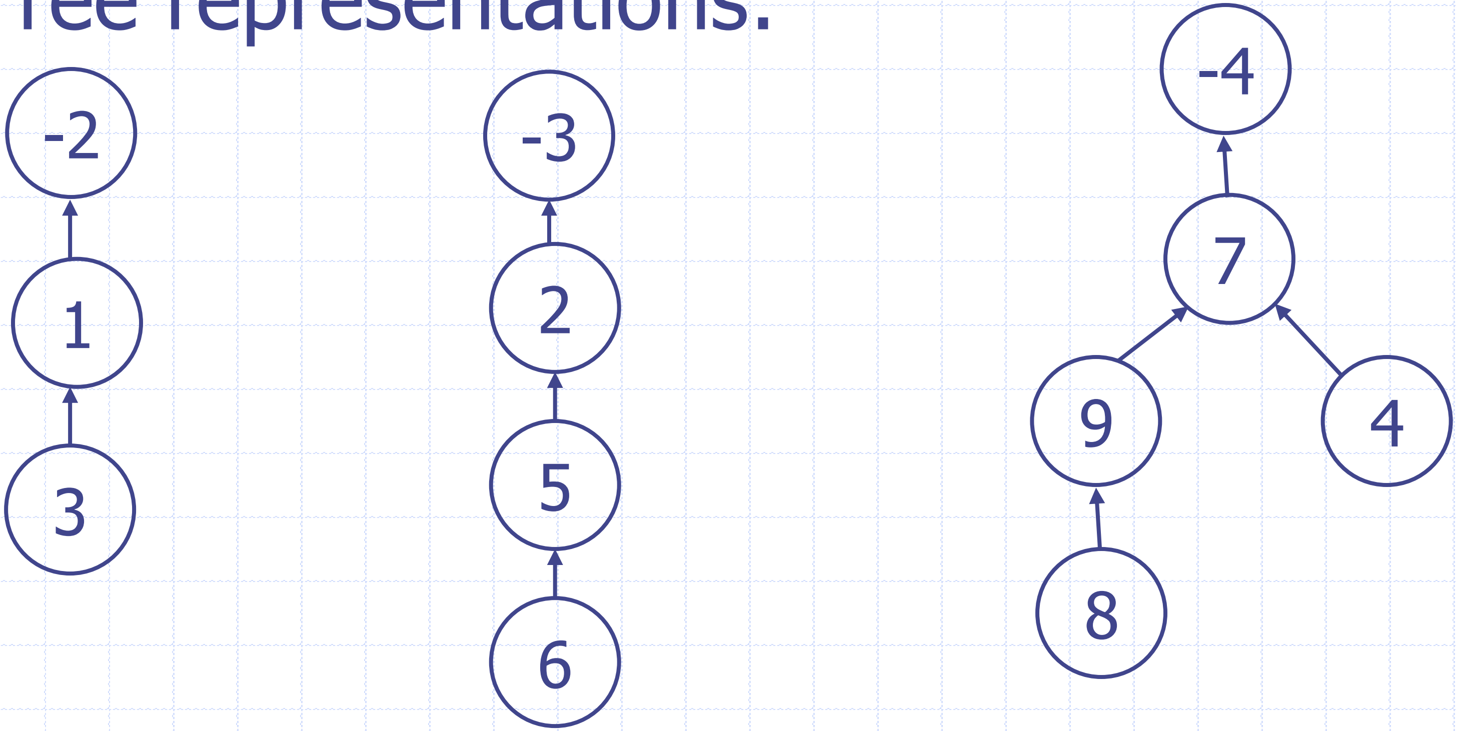
Array Implementation

1	2	3	4	5	6	7	8	9
-2	-3	1	7	2	5	-4	9	7

U = {1, 2, 3, 4, 5, 6, 7, 8, 9}

C = {{1, 3}, {2,5,6}, {4, 7, 8, 9}}

Tree representations:



Example

U = {1, 2, 3, 4, 5, 6, 7, 8, 9}

C = {{1, 3}, {2,5,6}, {4, 7, 8, 9}}

Find(1) = Find(3) = 1. (The root)

Arr[1] = -2. Here 2 denote number of items. Also, the negative sign indicates end of the tree.

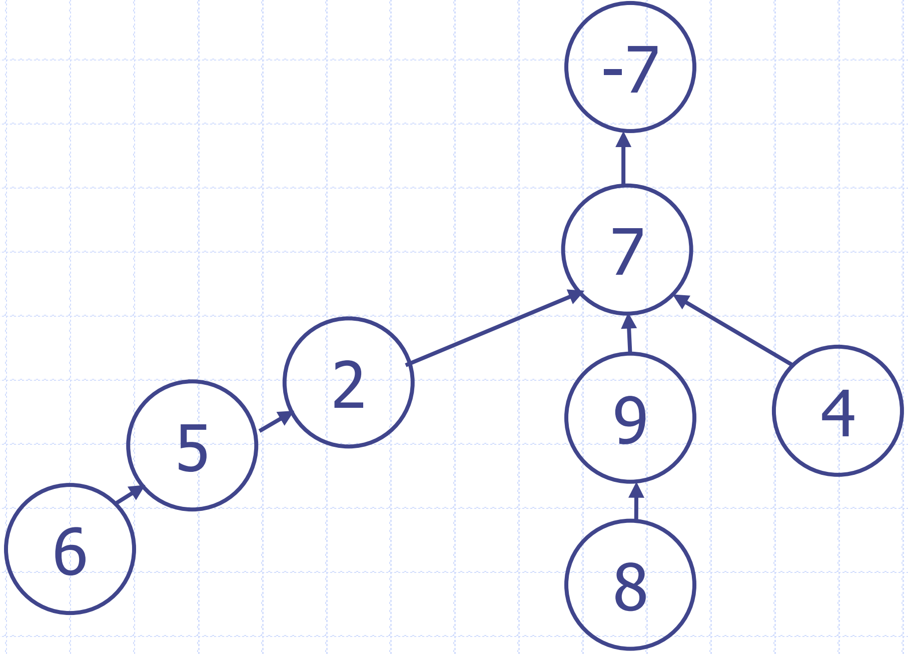
Find(2) = Find(5) = Find(6) = 2

Find(4) = Find(7) = Find(8) = Find(9) = 7

1	2	3	4	5	6	7	8	9
-2	7	1	7	2	5	-7	9	7

Union(6, 8)

- Make the “root” of the smaller tree point to root of larger tree.
- Change the value of number of items



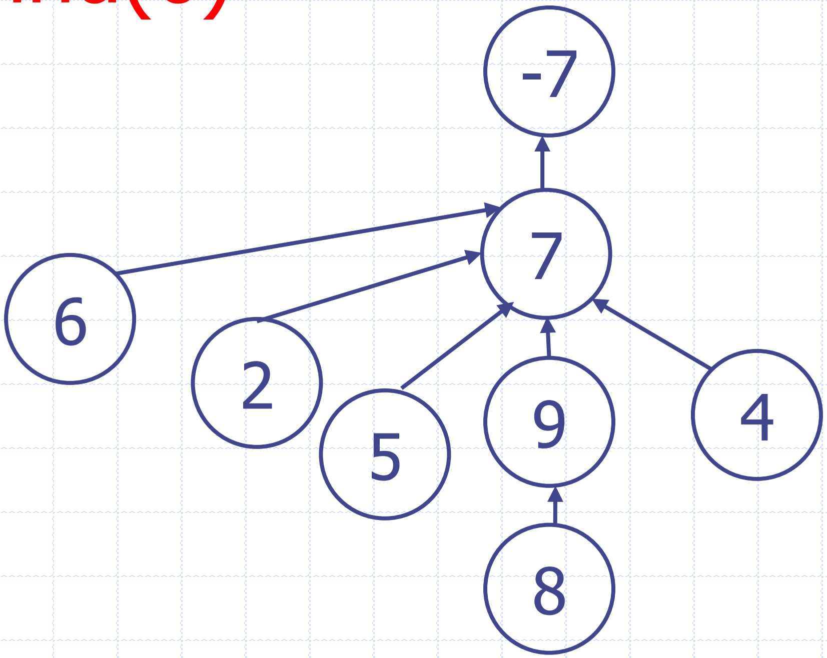
Find(5) is 7

1	2	3	4	5	6	7	8	9
-2	7	1	7	7	7	-7	9	7

Find with path compression

All elements from the “findItem” to the root of the tree will point to the root of the tree.

Find(6)



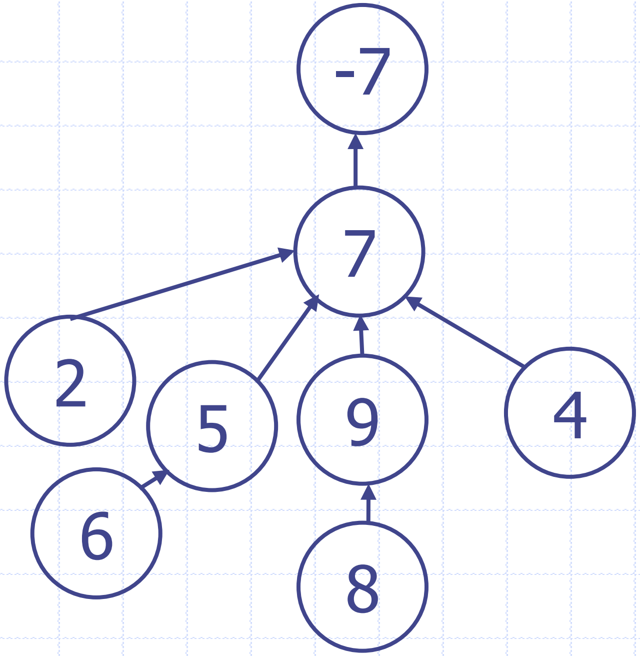
Note that 6 and 5 pointing to 7.

1	2	3	4	5	6	7	8	9
-2	7	1	7	7	5	-7	9	7

Find with path compression

All elements from the “find item” to the root of the tree will point to the root of the tree.

Find(5) (Instead of Find(6))



Note that 6 still points to 5 since 6 was “before” 5.

Summary

Thus, the disjoint sets are implemented using an 1-dimensional array. With path compression, we can almost achieve O(1) time complexity for union and find operations.

In this example, index 0 is not used. If elements in the sets are 0, 1, …, then we can use index location 0 as well.