Disjoint Tree: a bunch of sets with no common elements

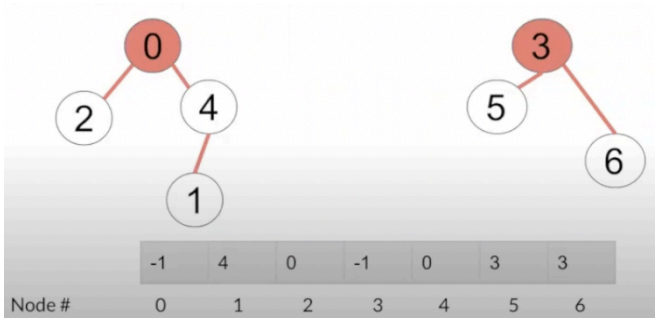
Union Find

1. Find: which of the set the given node belongs to
2. Union: joining to 2 sets into 1 set
    a. Void connect(x, y): connects (set of) node x and (set of) node y
    b. isConnected(x, y): returns whether or not there is a path between x and y
        i. If they're equal to each other then it's true
        ii. If Find(x) and Find(y) belong to the same set

Quick Union

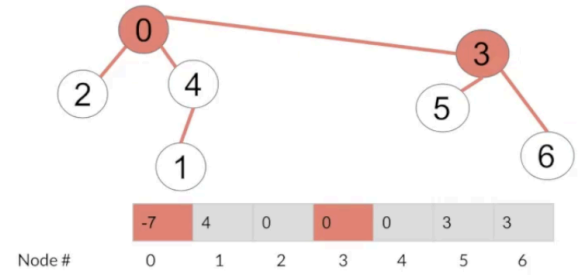
1. Use an int[] to represent the parent-child relationship between nodes
2. For each node, array[node] = parent node
3. Array[root node] = - 1

Worst case when calling find: O(N) when the union is spindly



Weighted Quick Union

4. Use an int[] to represent the parent-child relationship between nodes
5. For each node, array[node] = parent node
6. Array[root node] = - 1 * size of the set



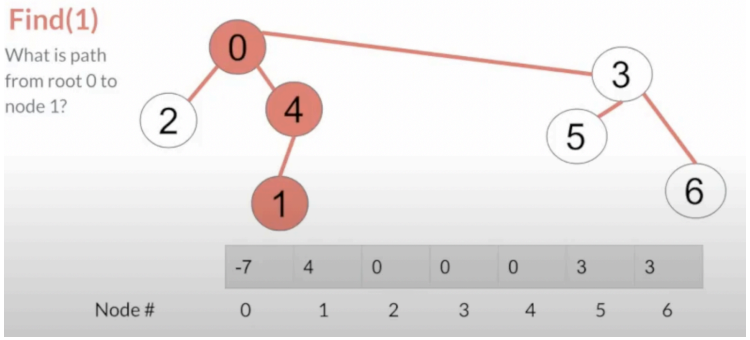
Worst case: O(log(N))

- Increase the height by 1 = double the size

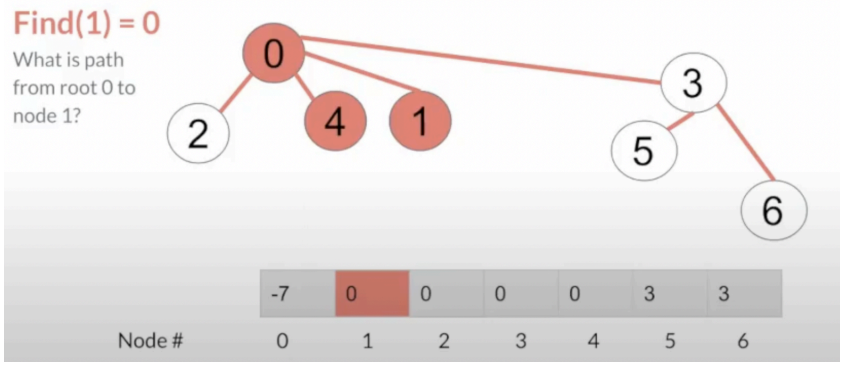
Weighted Quick Union with Path Compression

Whenever we use **Find** operation, **compress** the path to the node (*connect every node that is on the path towards the node* to root node)

- Make every node on the path a direct child of the root node
- Decreases max height and makes it constant time operation



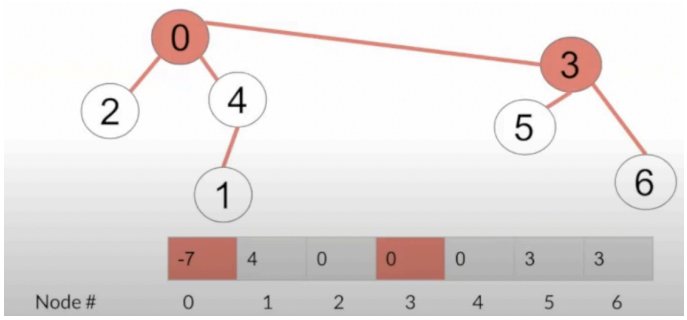
parent[1] = find(parent[1])



Implementation	Constructor	connect()	isConnected()
QuickUnion	Θ(N)	O(N)	O(N)
QuickFind	Θ(N)	O(N)	O(1)
Weighted Quick Union	Θ(N)	O(log N)	O(log N)
WQU with Path Compression	Θ(N)	O(log N) Θ(1)*	O(log N) Θ(1)*

When you call connect(x, y) with sets, you want to connect them together at their roots

- **Example: connect(0, 3)**



When you connect sets of a weighted quick union…

1. Max height should be bonded by O(log(N))
2. Must nest smallest sets under larger ones
3. You go up the path until you reach a negative number (aka the root)
4. You make the greater negative number (smaller size tree) the child of the other root
5. The other root size changes to be the size of the two sets combined

- You go from largest path to smallest path
- Connect the bigger tree first so that the other tree goes underneath the root