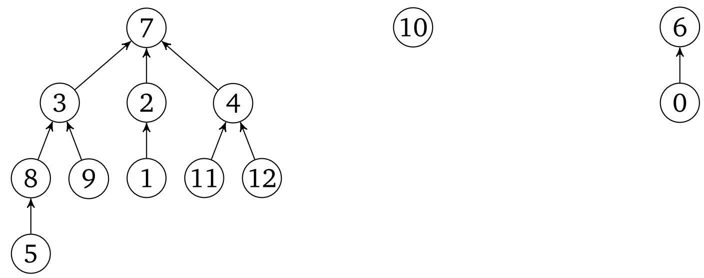
Section 09: Solutions

1. Disjoint sets

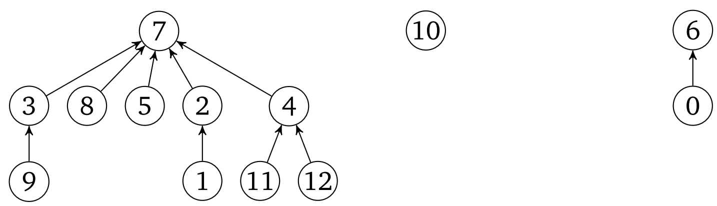
(a) Consider the following trees, which are a part of a disjoint set data-structure:



For these problems, use both the **weighted quick union by size** and **path compression** optimizations.

(i) Draw the resulting tree(s) after calling `find(5)` on the above. What value does the method return?

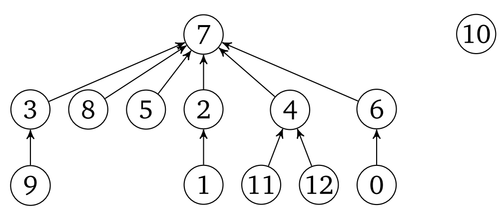
Solution:



The method returns 7, which is the root of the tree.

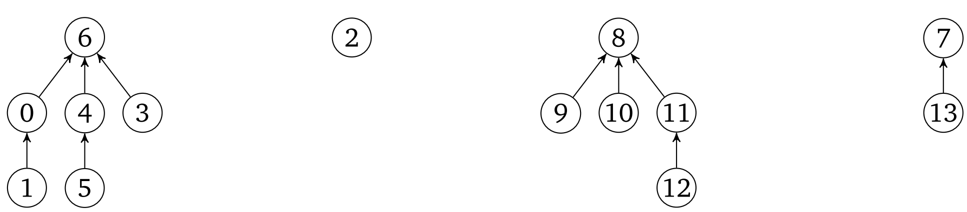
(ii) Draw the final result of calling `union(2,6)` on the result of part a.

Solution:



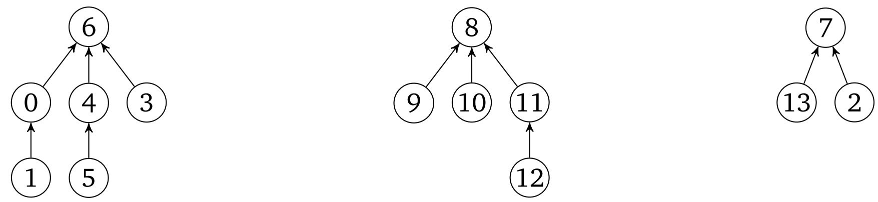
We merge the second tree to the first since the size of the first tree is larger.

(b) Consider the disjoint-set shown below

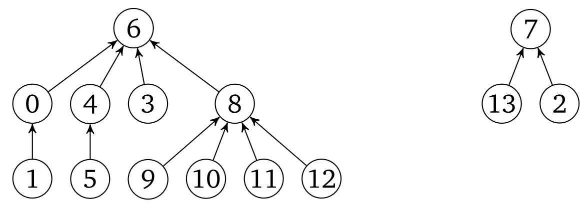


What would be the result of the following calls on union if we add the "weighted quick union by size" and "path compression optimizations.

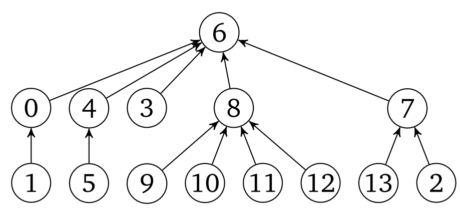
(i) `union(2, 13)` **Solution:**



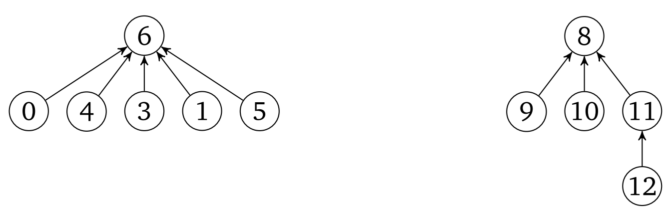
(ii) `union(4, 12)` **Solution:**



(iii) `union(2, 8)` **Solution:**

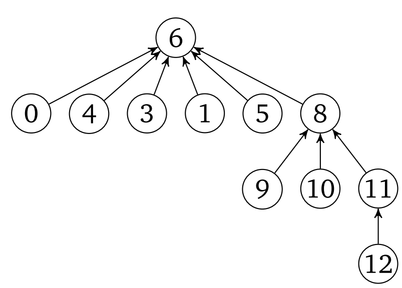


(c) Consider the disjoint-set shown below



What would be the result of the following calls on union if we add the "weighted quick union by size" and "path compression optimizations.

(i) `union(10, 0)` **Solution:**



Note in particular here that if we do the weighted quick union by size we are always choosing the tree with the larger number of nodes to adopt the tree with the smaller number of nodes. Note that we want to keep the height as small as possible, so in this case, the “best solution” would have been to connect the 6 to the 8, since the resulting height would be 2 instead of 3. However, we can’t know the exact height of these trees when we are performing the path compression optimization, so we accept sub-par results like this in some cases. It’s easier to always connect the tree with the smaller number of nodes to the tree with the larger number of nodes than it is to calculate the exact height of both trees (note that it would take $O(n)$ time to calculate the height of both trees, but we need this operation to be fast).