14.4 Weighted Quick Union (WQU)

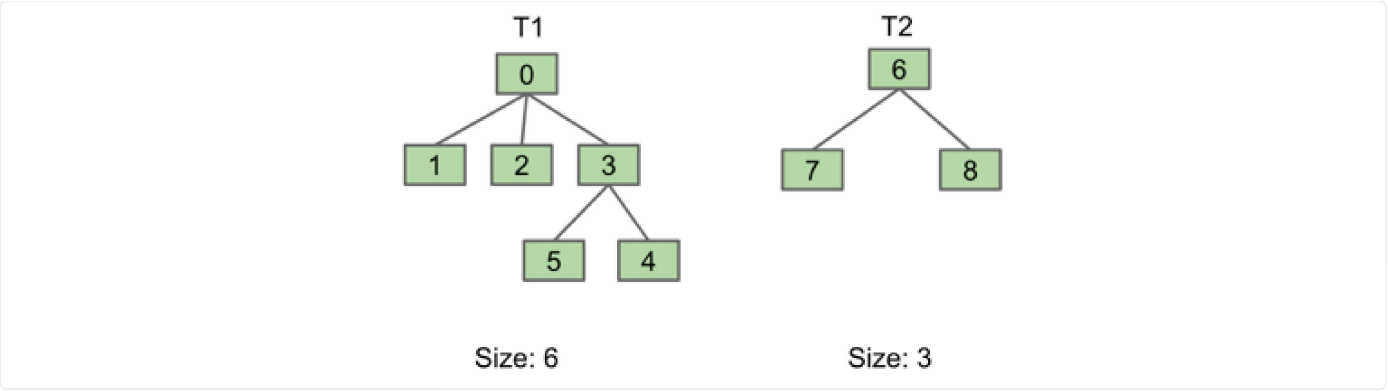
Improving on Quick Union relies on a key insight: whenever we call `find`, we have to climb to the root of a tree. Thus, the shorter the tree the faster it takes!

New rule: whenever we call `connect`, we always link the root of the smaller tree to the larger tree.

Following this rule will give your trees a maximum height of $logN$, where N is the number of elements in our Disjoint Sets. How does this affect the runtime of `connect` and `isConnected`?

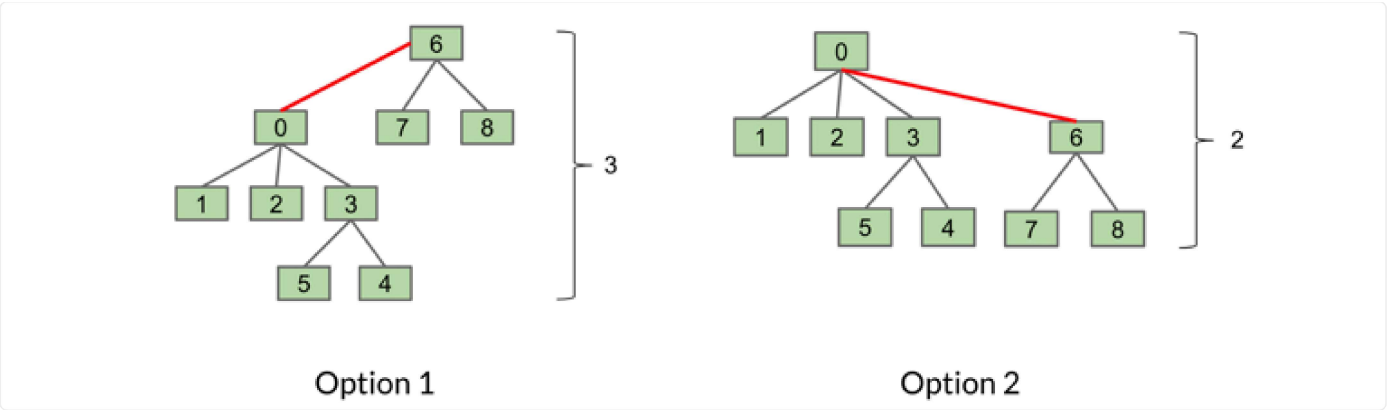
Professor Hug's explanation on Weighted Quick Union

Let's illustrate the benefit of this with an example. Consider connecting the two sets T1 and T2 below:



We have two options for connecting them:

The first option we link T1 to T2. In the second, we link T2 to T1.



The **second option is preferable** as it only has a height of 2, rather than 3. By our new rule, we would choose the second option as well because T2 is smaller than T1 (size of 3 compared to 6).

We determine smaller / larger by the number of items in a tree. Thus, when connecting two trees we need to know their size (or weight). We can store this information in the root of the tree by replacing the `-1`'s with `-(size of tree)`.

Maximum height: Log N

Following the above rule ensures that the *maximum* height of any tree is Θ(log N). N is the number of elements in our Disjoint Sets. **By extension, the runtimes of `connect` and `isConnected` are bounded by O(log N).**

Why $logN$? The video above presents a more visual explanation. Here's an optional mathematical explanation why the maximum height is $log_2 N$. Imagine any element x in tree $T1$. The depth of x increases by 1 only when $T1$ is placed below another tree $T2$. When

that happens, the size of the resulting tree will be at least double the size of $T1$ because $size(T2) \geq size(T1)$. The tree with x can double at most $log_2 N$ times until we've reached a total of N items ($2^{log_2 N} = N$). So we can double up to $log_2 N$ times and each time, our tree adds a level $\rightarrow$ maximum $log_2 N$ levels.

You may be wondering why we don't link trees based off of height instead of weight. It turns out this is more complicated to implement and gives us the same Θ(log N) height limit.

Summary

Implementation	Constructor	`connect`	`isConnected`
QuickUnion	Θ(N)	O(N)	O(N)
QuickFind	Θ(N)	Θ(N)	Θ(1)
QuickUnion	Θ(N)	O(N)	O(N)
Weighted Quick Union	Θ(N)	O(log N)	O(log N)

N = number of elements in our DisjointSets data structure

Previous
14.3 Quick Union

Next
14.5 Weighted Quick Union with Path Compression

Last updated 1 year ago