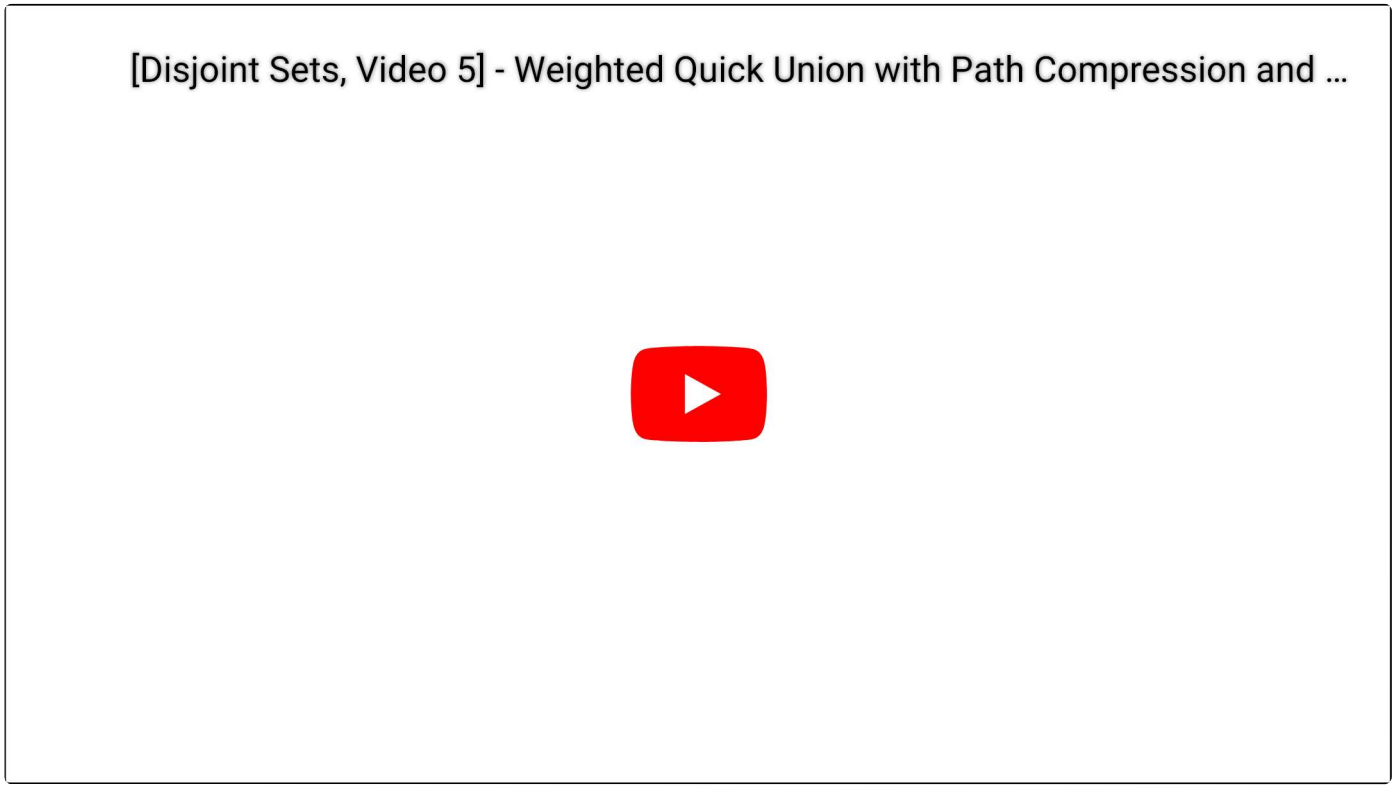
14.5 Weighted Quick Union with Path Compression

Weighted Quick Union is pretty good, but we can do even better!



Professor Hug's explanation on Weighted Quick Union with Path Compression

The clever insight is realizing that whenever we call `find(x)` we have to traverse the path from `x` to root. So, along the way we can connect all the items we visit to their root at no extra asymptotic cost.

Connecting all the items along the way to the root will help make our tree shorter with each call to `find`.

Recall that **both `connect(x, y)` and `isConnected(x, y)` always call `find(x)` and `find(y)`**. Thus, after calling `connect` or `isConnected` enough, essentially all elements will point directly to their root.

By extension, the average runtime of `connect` and `isConnected` becomes **almost constant** in the long term! This is called the *amortized runtime*.

More specifically, for M operations on N elements, WQU with Path Compression is in $O(N + M(lg * N))$. lg* is the iterated logarithm which is less than 5 for any real-world input.

Summary

N: number of elements in Disjoint Set

Implementation	`isConnected`	`connect`
Quick Find	Θ(1)	Θ(N)
Quick Union	O(N)	O(N)
Weighted Quick Union (WQU)	O(log N)	O(log N)
WQU with Path Compression	O(α(N))*	O(α(N))*

*behaves as constant in long term.

Previous
14.4 Weighted Quick Union (WQU)

Next
14.6 Exercises

Last updated 1 year ago