Back to Chapter

Disjoint Set

- Overview of Disjoint Set
- Quick Find - Disjoint Set
- Quick Union - Disjoint Set
- Union by Rank - Disjoint Set
- Path Compression Optimiz...
- Optimized "disjoint set" wi...
- Summary of the "disjoint s...
- Number of Provinces

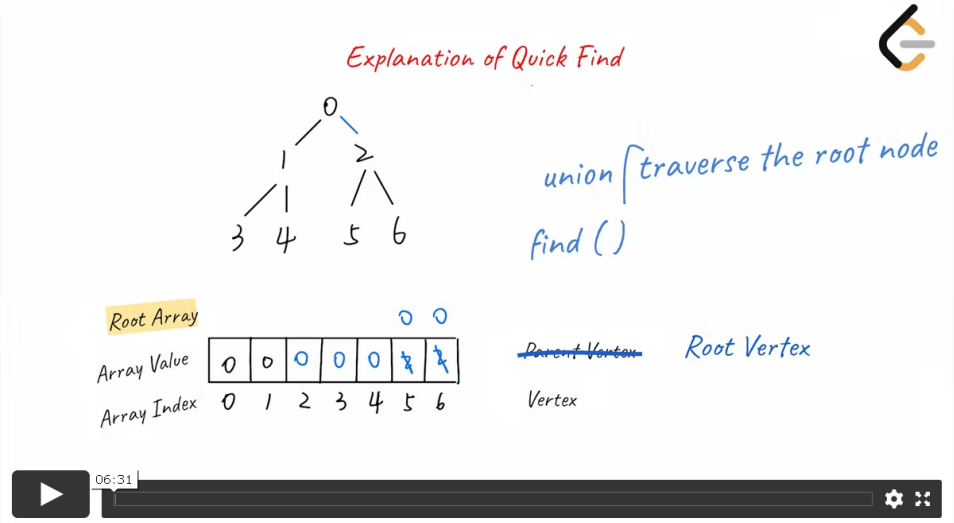
Previous Next

Quick Find - Disjoint Set

Report Issue

Explanation of Quick Find

In this video, we'll talk about Quick Find implementation of a Disjoint Set and cover its two basic operations along with their complexity: find and union.



Algorithm

Here is a sample quick find implementation of the Disjoint Set.

C++ Java Python3 Copy Run Playground

```cpp
class UnionFind {
public:
    UnionFind(int sz) : root(sz) {
        for (int i = 0; i < sz; i++) {
            root[i] = i;
        }
    }

    int find(int x) {
        return root[x];
    }

    void unionSet(int x, int y) {
        int rootX = find(x);
        int rootY = find(y);
        if (rootX != rootY) {
            for (int i = 0; i < root.size(); i++) {
                if (root[i] == rootY) {
                    root[i] = rootX;
                }
            }
        }
    }

    bool connected(int x, int y) {
        return find(x) == find(y);
    }
```

Time Complexity

	Union-find Constructor	Find	Union	Connected
Time Complexity	$O(N)$	$O(1)$	$O(N)$	$O(1)$

Note: N is the number of vertices in the graph.

- When initializing a `union-find constructor`, we need to create an array of size N with the values equal to the corresponding array indices; this requires linear time.
- Each call to `find` will require $O(1)$ time since we are just accessing an element of the array at the given index.
- Each call to `union` will require $O(N)$ time because we need to traverse through the entire array and update the root vertices for all the vertices of the set that is going to be merged into another set.
- The `connected` operation takes $O(1)$ time since it involves the two `find` calls and the equality check operation.

Space Complexity

We need $O(N)$ space to store the array of size N.