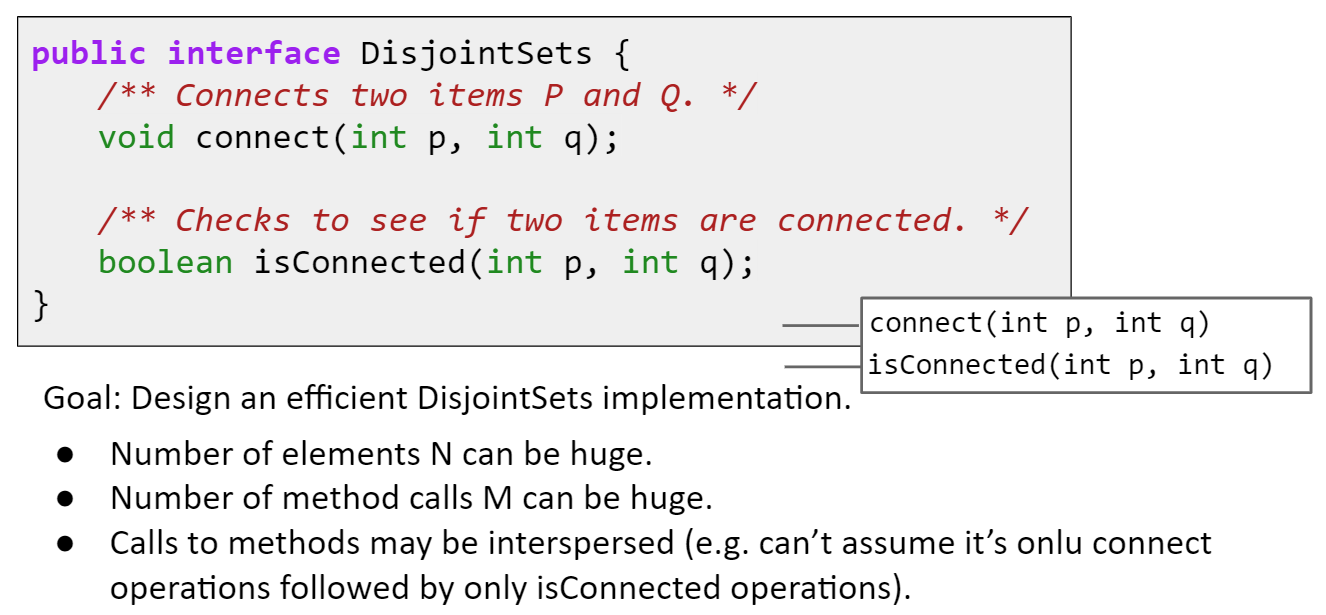
Disjoin Sets:

**Usefulness:**

* Percolation theory:
* Implementation of other algorithms:

The Disjoint Sets Interface



**Quick Find:**

List of integers where ith entry gives set number of item i.

Connect(p, q): change entries that equal id[p] to id[q].

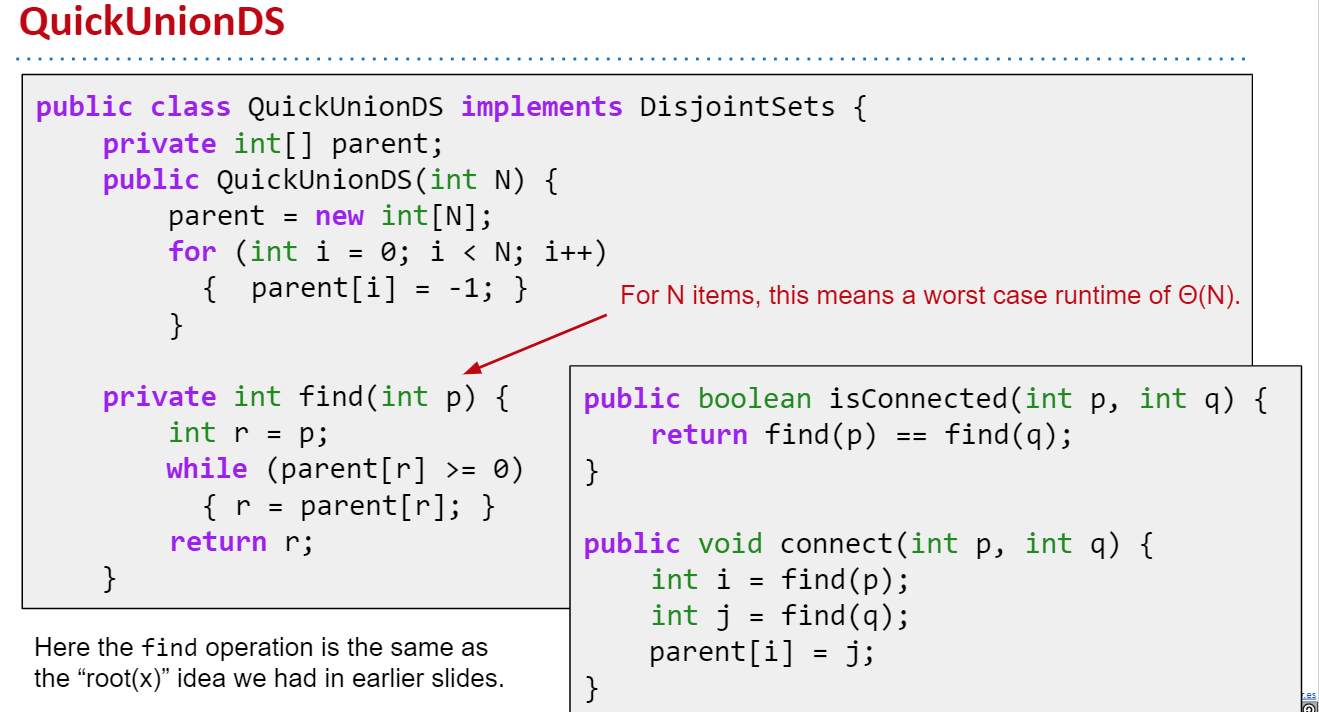
|  |  |  |  |
| --- | --- | --- | --- |
| Implementation | Constructor | Connect | isConnected |
| QuickFindDS | Ɵ(N) | Ɵ(N) | Ɵ(1) |
| QuickUnionDS | Ɵ(N) | O(N) | O(N) |
| WeightedQuickUnionDS | Ɵ(N) | O (log N) | O (log N) |
|  |  |  |  |

**Quick Union:**

Improving connect operation: Assigning each item a parent. Results in a tree like shape.

Connect the new item to the root.

**Potential problem:** May result in a linked list like data structure.



**Weighted Quick Union:**

Modify quick-union to avoid tall trees.

* Track tree size (**number** of elements).
* New rule: Always link root of ***smaller*** tree ***to*** ***larger*** tree.

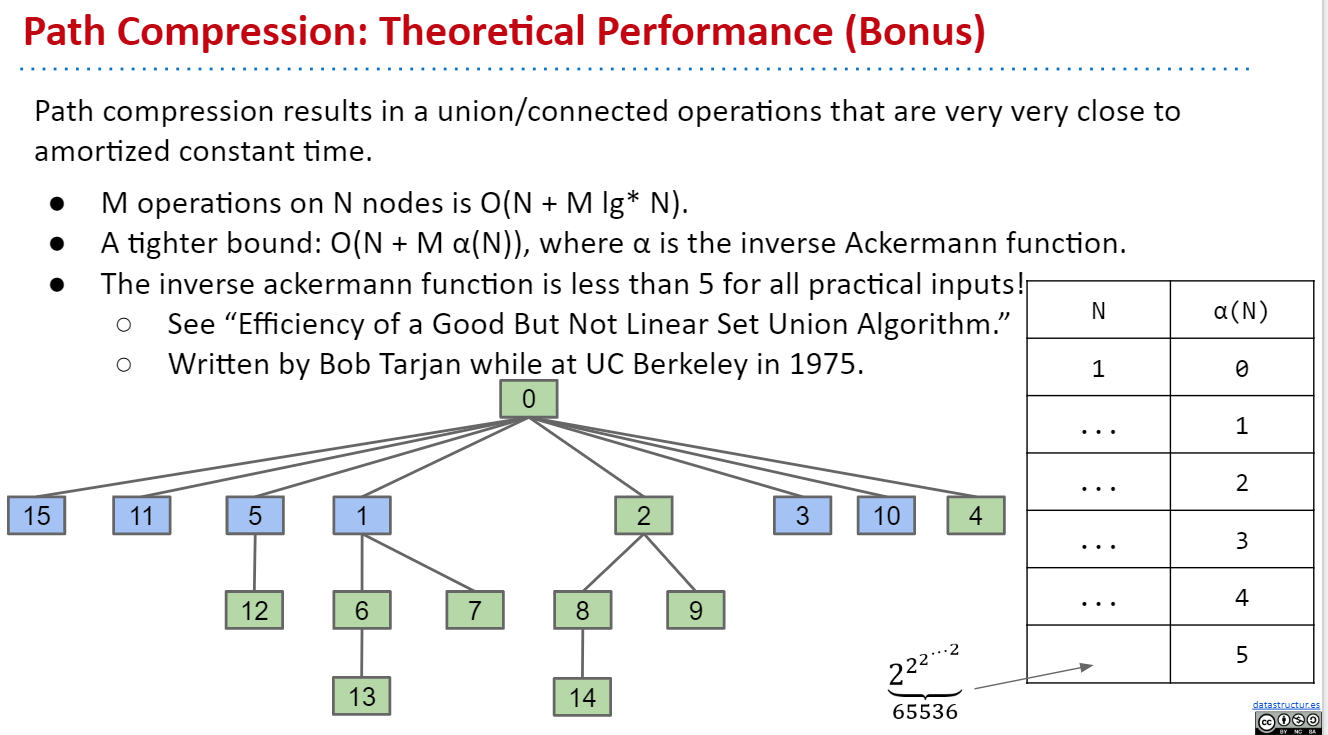
connect(int p, int q) needs to somehow keep track of sizes:

* Use values other than -1 in parent array for root nodes to track size.
* Create a separate size array.

Worst case tree height is Ɵ (log N). QuickUnion’s runtimes are O(H), and WeightedQuickUnionDS height is given by H = O(log N). Therefore connect and isConnected are both O(log N).

**Path Compression:**

 When we do isConnected(int p, int q), tie all nodes seen to the root.



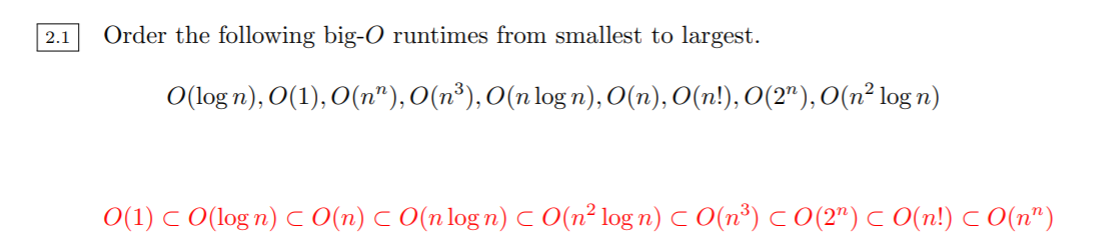
log\*(n) is the iterated log - it’s the number of times you need to apply log to n to go below 1. Note that 2^65536 is higher than the number of atoms in the universe.

|  |  |
| --- | --- |
| Implementation | Runtime |
| ListOfSetsDS | O(NM) |
| QuickFindDS | Θ(NM) |
| QuickUnionDS | O(NM) |
| WeightedQuickUnionDS | O(N + M log N) |
| WeightedQuickUnionDSWithPathCompression | O(N + M α(N)) |

**Disc06:**

* 1. (a) keep track of the tree size.

(b) path compression.



注意 n!在2^n和n^n之间。