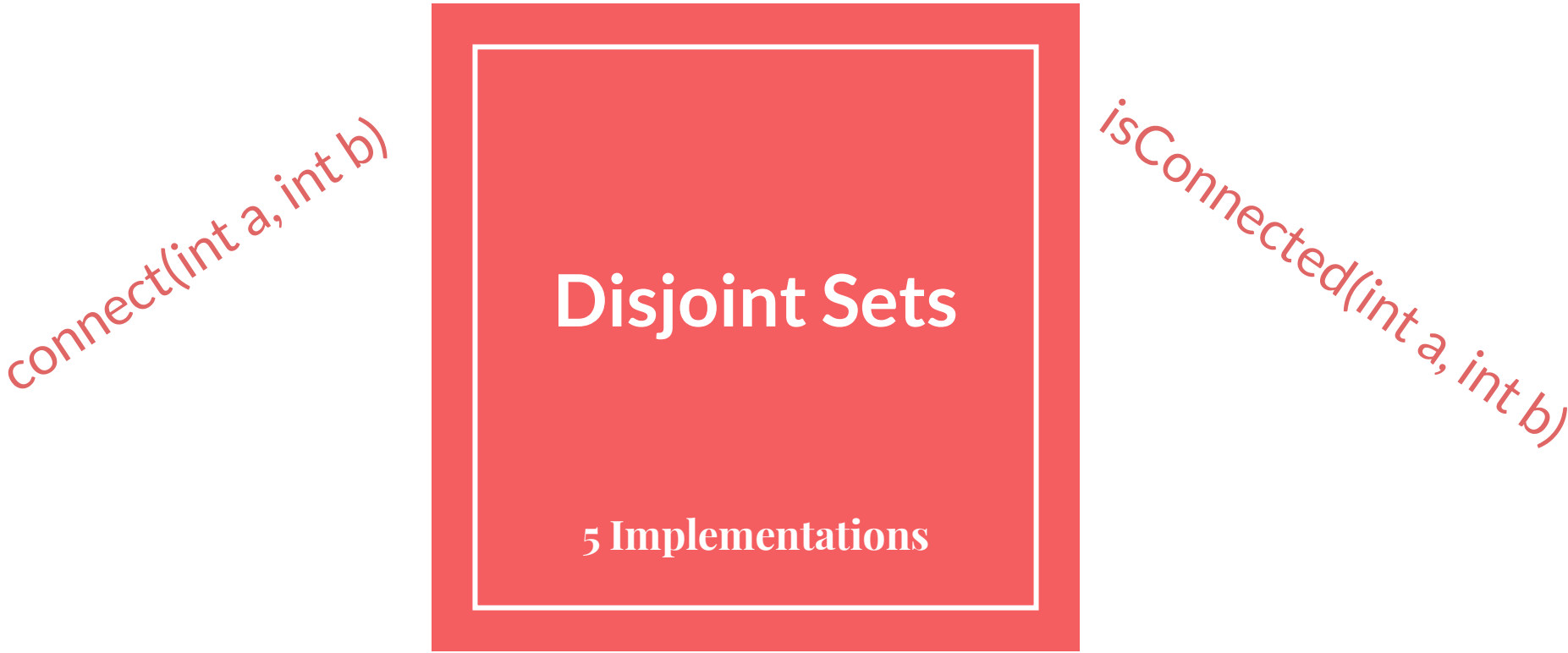
connect(int a, int b)
Disjoint Sets
5 Implementations
isConnected(int a, int b)

What > How

Name	Runtime			Key Characteristic	Pros	Cons
	Construc tor	connect()	isConnected()			
List of Sets	**Θ(N)**	**O(N)**	**O(N)**	-Connected item in same basket	-Focused on connected component	-Worst case have slow operations
Quick Find	**Θ(N)**	**Θ(N)**	**Θ(1)**	-stores setID	-quick isConnected()	-slow connect() (iterate through whole array)
Quick Union	**Θ(N)**	**O(N)**	**O(N)**	-stores parent item -root item: negative value	-quick connect()	-expensive for tall trees
Weighted Quick Union	**Θ(N)**	**O(log N)**	**O(log N)**	-connects smaller set to larger set	-worst case height log N	
Weighted Quick Union with Path Compression	**Θ(N)**	**O(lg* N)**	**O(lg* N)**	-tying all nodes to the root if seen	-tighter bounds than WQU	

Key Takeaway

1. Care about WHAT are connected not HOW
2. Choice of underlying abstraction impacts runtime & code complexity
    a. Edges vs. List of sets vs. Array
3. Algorithm development is an iterative process
    a. Tweaking existing model to greatly improve performance