

# Search Tree implementations: summary

implementation	guarantee			average case			ordered ops?	operation on keys
	search	insert	delete	search hit	insert	delete		
sequential search (unordered list)	N	N	N	N/2	N	N/2	no	equals()
binary search (ordered array)	lg N	N	N	lg N	N/2	N/2	yes	compareTo()
Binary Search Tree	N	N	N	1.39 lg N	1.39 lg N	sqrt(N)	yes	compareTo()
2-3 Tree	c lg N	c lg N	c lg N	c lg N	c lg N	c lg N	yes	compareTo()
red-black BST	2 lg N	2 lg N	2 lg N	1.00 lg N*	1.00 lg N*	1.00 lg N*	yes	compareTo()

- exact value of coefficient unknown but extremely close to 1.

## 2-3 tree: implementation

Direct implementation is complicated because:

- Maintaining multiple node types is cumbersome
- Need multiple compares to move down tree.
- Need to move back up the tree to split 4-nodes
- large number of cases for splitting

Bottom line. Could do it, but there is a better way.