Summary Tables

These are all the tables summarizing the comparisons between different data structures and algorithms covered in this course.

Access and Modification Characteristics

	get/set	**add/remove**
Arrays	*O(1)*	*O(1 + min(i,n-i))*
LinkedList	*O(1 + min(i,n-i))*	*O(1)**
Skiplist	*O(log n)*	*O(log n)*

`*given a pointer to the location, else traversal is necessary`

Binary Search Tree Implementations

	find()	**add()**	**remove()**
BST	*O(n)*	*O(n)*	*O(n)*
RBST / Treaps	*O(log n)* [expected]	*O(log n)* [expected]	*O(log n)* [expected]
Scapegoat Trees	*O(log n)* [amortized]	*O(log n)* [amortized]	*O(log n)* [amortized]
2-4 / RedBlack Trees	*O(log n)* [worst-case]	*O(log n)* [worst-case]	*O(log n)* [worst-case]

Sorted Set Implementations

	Runtime

Skiplists	*O(log n)* [expected]
Treaps	*O(log n)* [expected]
Scapegoat Trees	*O(log n)* [amortized]
2-4 / RedBlack Trees	*O(log n)* [worst-case]

Comparison-based Algorithms

	Comparisons	In-place	Stable
Merge Sort	*n•log(n)* [worst-case]	no	yes
Heap Sort	*1.38n•log(n) + O(n)* [expected]	yes	no
Quick Sort	*2n•log(n) + O(n)* [worst-case]	yes	no

Graph Implementations

	Adjacency Matrix	Adjacency List
addEdge	*O(1)*	*O(1)*
removeEdge	*O(1)*	*O(deg(i))*
hasEdge	*O(1)*	*O(deg(i))*
outEdge	*O(n)*	*O(1)*
inEdge	*O(n)*	*O(n+m)*
space used	*O(n^2)*	*O(n+m)*

Adjacency Matrix vs. Adjacency List

It is better to use **Adjacency List** for **traversals**.

	Adjacency Matrix	Adjacency List
Breadth	*O(n^2)*	*O(n+m)*

Depth	*O(n^2)*	*O(n+m)*