

Abstract Data Types and Data Structures

Summary

ADT	Operation	Type	Best Data Structure	Big O	Alt data Structure	Big O
List (Sequence)	Get	R	Dynamic Array	$O(1)$	Linked List	$O(n)$
	Add	W	Linked List	$O(1)$ ^{when the place is known}	Dynamic Array	$O(n)$
	Remove	W	Linked List	$O(1)$ ^{when the place is known}	Dynamic Array	$O(n)$
Stack (LIFO queue)	Top	R	Linked List	$O(1)$	Dynamic Array	$O(1)$
	Pop	W	Linked List	$O(1)$	Dynamic Array	$O(1)$
	Push	W	Linked List	$O(1)$	Dynamic Array	$O(1)$
Queue (FIFO queue)	Top	R	Linked List	$O(1)$	Circular Dynamic Array	$O(1)$
	Enqueue	W	Linked List ^{with tail}	$O(1)$	Circular Dynamic Array	$O(1)$ ^{amortized}
	Dequeue	W	Linked List	$O(1)$	Circular Dynamic Array	$O(1)$
Fast Local Search	Find	R	Balanced BST	$O(\log n)$	Sorted Array	$O(\log n)$
	Neighbor	R	Balanced BST	$O(\log n)$	Sorted Array	$O(\log n)$
	Add	W	Balanced BST	$O(\log n)$	Linked List	$O(1)$
	Remove	W	Balanced BST	$O(\log n)$	Linked List	$O(1)$ ^{when the place is known}
Set	Haskey	R	Hash Table	$O(1)$ ^{expected}	Balanced BST	$O(\log n)$

	Add	W	Hash Table	$O(1)^{\text{expected}}$	Balanced BST	$O(\log n)$
	Remove	W	Hash Table	$O(1)^{\text{expected}}$	Balanced BST	$O(\log n)$
Map	Get	R	Hash Table	$O(1)^{\text{expected}}$	Balanced BST	$O(\log n)$
	Put	W	Hash Table	$O(1)^{\text{expected}}$	Balanced BST	$O(\log n)$
	Remove	W	Hash Table	$O(1)^{\text{expected}}$	Balanced BST	$O(\log n)$
Priority Queue	Top	R	Binary Heap	$O(1)$	Balanced BST	$O(\log n)$
	Enqueue	W	Binary Heap	$O(\log n)$	Balanced BST	$O(\log n)$
	Dequeue	W	Binary Heap	$O(\log n)$	Balanced BST	$O(\log n)$
Graph	GetNeighbors	R	Adjacency Lists	$O(1)$	Adjacency Matrix	$O(n)$
	HasEdge	R	Adjacency Matrix	$O(1)$	Adjacency Lists	$O(n)$
	AreAdjacent	R	Adjacency Matrix	$O(1)$	Adjacency Lists	$O(n)$
	AddEdge	W	Adjacency Lists	$O(1)$	Adjacency Matrix	$O(1)$
	AddVertex	W	Adjacency Lists	$O(1)$	Adjacency Matrix	$O(n)$
	RemoveEdge	W	Adjacency Matrix	$O(1)$	Adjacency Lists	$O(n)$
	RemoveVertex	W	Adjacency Lists	$O(1)$	Adjacency Matrix	$O(n)$