Complexity Summary

by Chris

2016年6月1日 12:21

Sorting Comparison https://en.wikipedia.org/wiki/Sorting_algorithm

Algorithm	In place	Stable	Input sensitive	Worst time	Avg time	Best time	Space
Bubble sort	+	+	Compare - Swap +	n^2	n^2	Compare n^2 Swap n	1
Selection sort	+	-	-	n^2	n^2	n^2	1
Insertion sort	+	+	+	n^2	n^2	n	1
Sell sort	+	-	+	Depends on Gap seq. Best known n(log n)^2	n(log n)^2 Or n^(1.5)	n	1
Merge sort	-	+	-	n log n	n log n	n log n	n
Quick sort	usually done in-place with O(log n) stack space	Typical in-place sort is not stable; stable versions exist	+	n^2 (already sorted)	n log n	n log n; variation is n	Average: log n , Worst: n; (stack space, related to tree height) Sedgewick variation is log n worst case.
Heap sort	+	-	-	n log n	n log n	n log n	1 if don't store result

Data structure comparison

	Worst access	Wost search	Worst insertion	Worst deletion	Avg access	Avg search	Avg insertion	Avg deletion	Best access	Best search	Best insertion	Best deletion	Space
unsorted Array	1	n	1	1	1	n	1	1	1	1	1	1	n
Sorted Array	1	n	n	n	1	n	n	n	1	1	n (find pos)	n (find pos)	n
Linked list	n	n	n	n	n	n	n	n	1	1	1	1	n
Stack	n	n	1	1	n	n	1	1	1	1	1	1	n
Hash table	-	n	n	n	-	1	1	1	-	1	1	1	n
BST	n (stick)	n (stick)	n	n	log n	log n	log n	log n	1	1	log n	log n (find buttom) + 1 Or 1 (find top) + log n (find to swap)	n
AVL tree	log n	log n	log n	log n	log n	log n	log n	log n	1	1	log n	log n (find buttom) + 1 Or 1 (find top) + log n (find to swap)	n

Search & select comparison

Algo	Worst time	Avg time	Best time
sequential Search	n	n	1
Binary search	log n	log n	1
Interpolation search	n	log log n	1
Quick select	n^2	n	n

(binary) Heap operation

Operation	Heapify	findMax	EjectMax	Inject	delete
Complexity	n (bottom up); n log n (top down)	1	log n	log n	n (find pos) + log n (re-heapify)