

Common Data Structure Operations									
Data Structure	Time Complexity					Space Complexity			
	Average	Search	Insertion	Deletion		Worst	Search	Insertion	Deletion
Array	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Stack	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Queue	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Singly-Linked List	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Doubly-Linked List	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Shred List	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Hash Table	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Bloom Filter	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
LRU Cache	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
AVL Tree	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$		$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Red-Black Tree	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$		$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Disjoint Set	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Trie	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Bitwise	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
Bloom Filter	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$
LRU Cache	$O(1)$	$O(1)$	$O(1)$	$O(1)$		$O(1)$	$O(1)$	$O(1)$	$O(1)$

空间复杂度

时间复杂度

最好

最坏

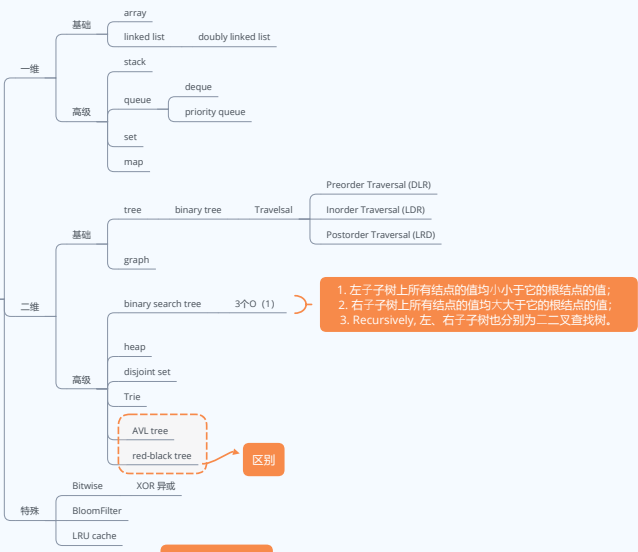
平均

均摊

复杂度分析

数据结构&算法

数据结构



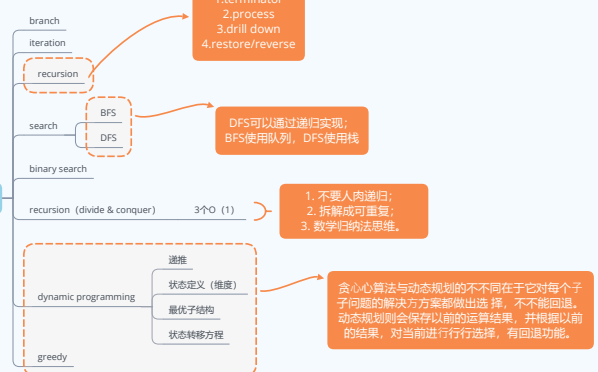
1. 左子子树上所有结点的值均小于它的根结点的值;
2. 右子子树上所有结点的值均大于它的根结点的值;
3. Recursively, 左、右子子树也分别为二叉查找树。

区别

- 1.terminator
- 2.process
- 3.drill down
- 4.restore/reverse

DFS可以通过递归实现;
BFS使用队列, DFS使用栈

算法



1. 不要人肉递归;
2. 拆解成可重复;
3. 数学归纳法思维。

贪心算法与动态规划的不同在于它对每个子问题的解决方案都做出选择, 不能回退。动态规划则会保存以前的运算结果, 并根据以前的结果, 对当前进行行选择, 有回退功能。