

## 数据结构时间复杂度总结

笔记本： Alvin's tech spec

创建时间： 2016/7/3 星期日 下午 5:55

更新时间： 2016/7/3 星期日 下午 5:56

作者： xlove\_wj@sina.com

URL： [http://mp.weixin.qq.com/s?\\_\\_biz=MzI2NjA3NTc4Ng==&mid=2652078199&id...](http://mp.weixin.qq.com/s?__biz=MzI2NjA3NTc4Ng==&mid=2652078199&id...)

### 数据结构操作

数据结构	时间复杂度								空间复杂度
	平均				最差				最差
	访问	搜索	插入	删除	访问	搜索	插入	删除	
<a href="#">Array</a>	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<a href="#">Stack</a>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<a href="#">Singly-Linked List</a>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<a href="#">Doubly-Linked List</a>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<a href="#">Skip List</a>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n \log(n))$
<a href="#">Hash Table</a>	-	$O(1)$	$O(1)$	$O(1)$	-	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<a href="#">Binary Search Tree</a>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<a href="#">Cartesian Tree</a>	-	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	-	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<a href="#">B-Tree</a>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<a href="#">Red-Black Tree</a>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<a href="#">Splay Tree</a>	-	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	-	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<a href="#">AVL Tree</a>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$

### 数组排序算法

算法	时间复杂度			空间复杂度
	最佳	平均	最差	最差
<a href="#">Quicksort</a> <sup>[14]</sup>	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$	$O(\log(n))$
<a href="#">Mergesort</a> <sup>[15]</sup>	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
<a href="#">Timsort</a> <sup>[16]</sup>	$O(n)$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
<a href="#">Heapsort</a> <sup>[17]</sup>	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(1)$
<a href="#">Bubble Sort</a> <sup>[18]</sup>	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
<a href="#">Insertion Sort</a> <sup>[19]</sup>	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
<a href="#">Selection Sort</a> <sup>[20]</sup>	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
<a href="#">Shell Sort</a> <sup>[21]</sup>	$O(n)$	$O((n \log(n))^2)$	$O((n \log(n))^2)$	$O(1)$
<a href="#">Bucket Sort</a> <sup>[22]</sup>	$O(n+k)$	$O(n+k)$	$O(n^2)$	$O(n)$
<a href="#">Radix Sort</a> <sup>[23]</sup>	$O(nk)$	$O(nk)$	$O(nk)$	$O(n+k)$

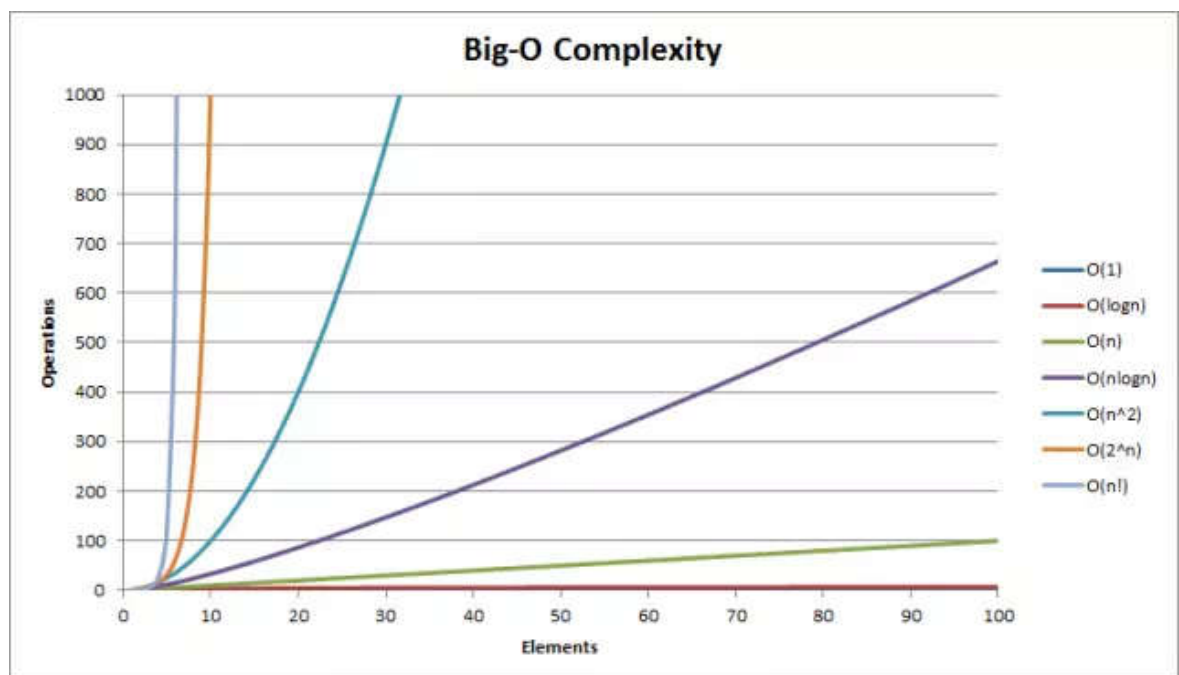
## 图操作

节点 / 边界管理	存储	增加顶点	增加边界	移除顶点	移除边界	查询
<a href="#">Adjacency list</a> <sup>[24]</sup>	$O( V + E )$	$O(1)$	$O(1)$	$O( V  +  E )$	$O( E )$	$O( V )$
<a href="#">Incidence list</a> <sup>[25]</sup>	$O( V + E )$	$O(1)$	$O(1)$	$O( E )$	$O( E )$	$O( E )$
<a href="#">Adjacency matrix</a> <sup>[26]</sup>	$O( V ^2)$	$O( V ^2)$	$O(1)$	$O( V ^2)$	$O(1)$	$O(1)$
<a href="#">Incidence matrix</a> <sup>[27]</sup>	$O( V  \cdot  E )$	$O( V  \cdot  E )$	$O( V  \cdot  E )$	$O( V  \cdot  E )$	$O( V  \cdot  E )$	$O( E )$

## 堆操作

类型	时间复杂度						
	Heapify	查找最大值	分离最大值	提升键	插入	删除	合并
<a href="#">Linked List (sorted)</a> <sup>[23]</sup>	-	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(m+n)$
<a href="#">Linked List (unsorted)</a> <sup>[20]</sup>	-	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
<a href="#">Binary Heap</a> <sup>[30]</sup>	$O(n)$	$O(1)$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(m+n)$
<a href="#">Binomial Heap</a> <sup>[31]</sup>	-	$O(1)$	$O(\log(n))$	$O(\log(n))$	$O(1)$	$O(\log(n))$	$O(\log(n))$
<a href="#">Fibonacci Heap</a> <sup>[32]</sup>	-	$O(1)$	$O(\log(n))$	$O(1)$	$O(1)$	$O(\log(n))$	$O(1)$

大 O 复杂度图表



Eig C 复杂度