链滴

Java - 数据结构

作者：someone33881

原文链接：https://ld246.com/article/1547221385939

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20180816.jpg?imageView2/1/w/960/h/520/interlace/1/q/100">
    本文主要是记录在学习 Java - 数据结构 过程中的一些知识点备
！
20190114
<h2 id="一-数据结构">一、数据结构</h2>
<h3 id="1-链表LinkedList">1、链表 LinkedList</h3>

链表即是由节点 Node 组成的线性集合，每个节点可以利用指针指向其他节点。它是一种包含了
个节点的、能够用于表示序列的数据结构
单向链表：链表中的节点仅指向下一个节点，并且最后一个节点指向空
双向链表：链表中的每个节点均具有两个指针 p、n,且 p 指向前驱节点&n 指向后继节点；
后一个节点的 n 指针指向 null
循环链表：每个节点指向下一个节点并且最后一个节点指向第一个节点的链表
？？时间复杂度？？：

索引：<code>O(n)</code>
搜索：<code>O(n)</code>
插入：<code>O(1)</code>
移除：<code>O(1)</code>

<h3 id="2-栈Stack">2、栈 Stack</h3>

栈是元素的集合，包含两个基本操作：push 操作可用于将元素压入栈，pop 操作可将栈顶元素
除
符合后入先出原则
？？时间复杂度？？：

索引：<code>O(n)</code>
搜索：<code>O(n)</code>
插入：<code>O(1)</code>
移除：<code>O(1)</code>

<h3 id="3-队列Queue">3、队列 Queue</h3>

队列是元素的集合，包含两个基本操作：enqueue 操作可用于将元素插入到队列中，dequeue
作则是将元素从队列中移除
符合先入先出原则
？？时间复杂度？？：

索引：<code>O(n)</code>
搜索：<code>O(n)</code>
插入：<code>O(1)</code>
移除：<code>O(1)</code>

<h3 id="4-树Tree">4、树 Tree</h3>

树是无向、连通的无环图

<h4 id="4-1-二叉树Binary-Tree">4.1、二叉树 Binary Tree</h4>

二叉树即是每个节点最多包含左子节点与右子节点这两个节点的树形数据结构
满二叉树：树中的每个节点仅包含 0 或 2 个节点
完美二叉树：二叉树中的每个叶节点都用于两个子节点，并且具有相同的高度
完全二叉树：除最后一层外，每一层上的节点数均达到最大值；在最后一层上只缺少右边的若干点

<h4 id="4-2-二叉搜索树Binary-Search-Tree">4.2、二叉搜索树 Binary Search Tree</h4>

二叉搜索树 BST 是一种特殊的二叉树，其任何节点中的值都会大于或等于其左子树中存储的值并且小于或者等于其右子树中存储的值
？？时间复杂度？？：

索引：<code>O(n)</code>
搜索：<code>O(n)</code>
插入：<code>O(n)</code>
移除：<code>O(n)</code>

<h4 id="4-3-字典树Trie">4.3、字典树 Trie</h4>

字典树，又称基数树或前缀树，能够用于存储键为字符串的动态集合或者关联数组的搜索树。树的节点并没有直接存储关联键值，而是该节点在树中的挂载位置决定了其关联键值。某个节点的所有节点都拥有相同的前缀，整棵树的根节点则是空字符串

20190113
4、Map
5、树

二叉树：每个节点最多只有两个分支的树结构
完全二叉树
满二叉树
堆
二叉查找树
平衡二叉树
红黑树
B-,B+,B*树
LSM 树

6、图
7、BFS DFS
20190111

队列 Queue

一种只允许在一端进行插入，在另一端进行删除的线性表结构。运行插入的一端叫队尾（rear）允许删除的一端叫队头（font）

支持 FIFO，尾部添加、头部删除（即先进先出）
队列分“单队列、循环队列”两种：单队列，即是“每次添加元素时，均是添加到队尾，存在假出的问题，也就是明明有位置却不能添加的情况”；循环队列，则是避免了假溢出的问题

假溢出：当尾部插入速度小于头部删除速度时，出现 rear==front 的现象，但是此时队列并没满，而且正好相反的是队列此时为空，存储空间最大，但继续插入元素时，rear 值已经到达 MAXSIZE 边界条件，此时无法插入也无法删除。
Java 数据结构之队列（Queue）
<ol start="2">
集合 Set

Set 继承于 Collection 接口，是一个不允许出现重复元素且无序的集合，主要有 HashSet 和 TreSet 两大实现类
在判断重复元素的时候，Set 集合通过调用 hashCode()和 equals()方法来实现

有序集合：集合里的元素可以根据 key 或 index 访问（List、Map）

无序集合：集合里的元素只能遍历（Set）

HashSet：哈希表结构，主要利用 HashMap 的 key 来存储元素，计算插入元素的 hashCode 获取元素在集合中的位置

TreeSet：红黑树结构，每一个元素都是树中的一个节点，插入的元素都会进行排序

<ol start="3">
列表 List

在 List 中，可以精确地控制列表中每个元素的插入位置，且可以通过整数索引（列表中的位置）问元素，搜索列表中的元素！不用于 Set,List 允许重复元素，且 List 是有序集合，Set 是无序集合</>

ArrayList：数组队列，也即动态数组；由数组实现，随机访问效率高，随机插入、随机删除效率
LinkedList：双向链表，可以被当做堆栈、队列或双向队列进行操作；LinkedList 随机访问效率，但是随机插入、随机删除效率高
Vector：矢量队列，也是一个动态数组；由数组实现，但是 Vetcor 是线程安全的（ArrayList 非程安全）
Stack：栈，继承于 Vector；FILO（先进后出）
