

20220328-算法&计算机网络

1.过程描述

2.结果输出

1.过程描述

1 散列表执行各种操作的时间都为 $O(1)$

	数组	链表	散列表（平均情况）	散列表（最糟情况）
3 读取	$O(1)$	$O(n)$	$O(1)$	$O(n)$
4 插入	$O(n)$	$O(1)$	$O(1)$	$O(n)$
5 删除	$O(n)$	$O(1)$	$O(1)$	$O(n)$

散列表使用数组来存储位置。使用散列表要避免最糟情况就要避免冲突（多个key映射到同一存储位置，通常会将冲突的key转化为链表的形式），而避免冲突需要：

1) 较低的填充因子

填充因子=散列表包含的元素数/位置总数。当填充因子大于0.7时，就要考虑调整散列表的长度

10 2) 良好的散列函数

良好的散列函数让数组中的值呈均匀分布

13 2. 队列是一种先进先出的数据结构，而栈是后进先出的数据结构。队列的操作有入队和出队

15 3. 广度优先搜索的运行时间为 $O(V+E)$ ，V为顶点数（将每个人添加到队列需要的时间是固定的），E为边数

17 4. 如果任务A依赖于任务B，在列表中任务A就必须在任务B后面，这成为拓扑排序

19 5. 树是一种特殊的图，它没有往后指的边

21 6. 最短路径

边无权重：广度优先搜索

边有权重：狄克斯特拉算法

边权重有负数：贝尔曼-福德算法

26 7. 贪婪算法

可以简单理解为每一步都是当前状况下的最优解（局部最优解），但整体来看不一定是全局最优解，是一种近似算法（用于解决NP完全问题）

29 8. 动态规划

动态规划可帮助在约束条件下找到最优解，当问题可分解为彼此独立且离散的子问题时，就可以使用动态规划来解决。（背包问题）

32 9. 二叉查找树

对于其中的每个节点，左子节点的值都比他小，右子节点则比他大
平均运行时间为 $O(\log n)$ ，最糟糕的情况下为 $O(n)$ 。

	数组	二叉查找树
36 查找	$O(\log n)$	$O(\log n)$
37 插入	$O(n)$	$O(\log n)$
38 删除	$O(n)$	$O(\log n)$

2.结果输出

今天主要把《图解算法》快速浏览了一遍，这原本不在计划之内，但因为介绍的东西比较入门，加上心里也比较迫切想窥一窥算法的门道，所以临时jam了进来。书上的内容总体而言确实很基础，有些东西之前也稍微有些了解（比如散列表、最短路径等等），所以阅读起来没什么难度，但有些算法并没有深入去思考内在的机理，也没动手用C++实现，所以只能说是囫圇吞枣勉强看完了。晚上主要看了计算机网络网络层的部分，相对而言都是一些概念性的知识，比较抽象枯燥，所以看起来速度很慢。明天争取把网络层、传输层以及应用层的内容看完，之后再用一天时间做习题检验一下学习成果。今天C++的实践不多，只用运算符重载和友元大概实现了向量的加法和因子乘。后面每天都得抽点时间写写程序，不然感觉很快就会变得稀疏。相信量变会引起质变。