数据结构学习总结

本次经过代码规范与数据结构的学习我了解到以下几点：

**一、泛型与集合**

1.泛型有点类似于object，但毕竟向下转型非常的不安全，为以免发生ClassCastException，所以引用了泛型，而其核心为，类在定义时，使用一个标记，此标记表示类中属性或方法参数的类型，例如List<String>即List对象里只能存放String类型，规定了存放的类型，同时<>里不能放基本类型，可以写为<? Extends>、<? Super>。

2.ArrayList在java基础篇有学过，就相当于一个动态数组，大小可变。而LinkedList是基于链表的。这两者在get和set方面，ArrayList比LinkedList要快，因为LinkedList只能找到每一个要匹配的之前的数，就得一个个地找。但如果是在add和remove方面，LinkedList会比ArrayList快，毕竟只需要找到之前那一个就够了，不需要去遍历。

3.HashMap和HashSet都是无序的，前者实现Map接口，后者实现Set接口，Map类中有自己的key和value，所以说前者会比后者在查找时快一点。

4.Collection接口是由两个子接口List、Set以及Map接口形成的，该接口中有add、clear、isEmpty、contains、remove等方法，而Collection接口在java.util包里。

**二、基础排序算法**

1.冒泡排序：就是在相邻的两个数中比较，先确定每个队列中最后一个，直到最后只剩下第一个时，就说明排序完成。要注意的是要设置好退出标记，否则会发生下标溢出。

2.归并排序：就是将一个队列先分成两组，再把这两组各自又分成两组，以此循环，直至每组都只有一个数时，将这些分开的组合起来进行排序，可通过递归完成。

3.快速排序：将第一个数作为标记对比，从队列最右边向左边扫描，找到比标记小的数，与标记对调，然后再从队列最左边向右边扫描，找到比标记大的数，重复，直到左边的指针和右边的指针重复，然后再把左边和右边各自重复以上操作，直到有序即可。

4.堆排序：最主要的的在于叶节点必须小于根节点，从而建造出堆，然后利用递归将最后的一个叶节点与最顶端的点交换，交换后就将下一个叶节点与最顶端的点进行交换，以此类推。难点在于左边的叶节点为2 × n + 1，右边的为2× n + 2，n = 总共需要排序的点的个数 ÷ 2 - 1。

5.桶排序：就相当于根据某个数的每一位，将其放在0 - 9号桶中的一个，然后再根据0 - 9号桶的顺序取出，直到所有数的所有位数都排完。基数排序是基于桶排序的，而以上就为基数排序。

三**、树、图的深度和广度搜索**

1.深度搜索：假设有条件1、2、3，那么当搜索到一个数满足条件1后对其进行条件2的判别，如果条件2不满足，就跳过该数，对下一个满足条件1的数，进行条件2判别，满足，则进行条件3判别。

2.广度搜索：同样是有3个条件，先将一个堆或树中满足条件1的挑出来，再在这里面挑出满足条件2的，然后再挑出满足条件3的数。

3.疑问：还是无法理解数据结构考核中第二题的“构建完成后用户可以输入一个点，利用图的搜索算法，判断此点是否在图中”一句话，我认为要搜索点是否在图中直接使用equals就好了，毕竟搜索某一点不需要同时满足那么多的条件，所以我使用了BFS进行了找到最短路径。

**四、代码规范**

1.进行了对命名、格式、语句、注释、异常等规范。

2.仍产生的问题：因为不太熟练，所以有些地方的规范仍然还是不太能记得住，所以之后一定要多敲多看。

**五、其他**

根据这段时间的自学，我发现我自己的自觉能力仍然不够，还不能够特别专注的进行学习。而且因为要学的有点多，所以有时候有点无法抓住重点。同时，我对文档的阅读能力实在是有点差，所以导致我对考核的题目的理解有点困难，以后会增强对这些方面的优化和精进。

17级08专业2班

王裴培

2018年7月26日