各种排序算法的时间复杂度



1、直接插入排序，因为该排序算法是先让左边有序，然后从右边依次选择数据放到左边，并使左边有序，那么最好的情况就是原序列本身就是升序，每次只需从右边往左边添加数据，不需要对左边已经排好序的数据进行排序，那么一共只需添加n次；最坏的情况是序列本身是降序的，那么每次从右边往左边添加数据的时候都要与左边的数据进行对比移动，第n次需要移动（n-1）次，则一共需要移动1,+2+3+....+(n-1)=n(n-1)/2次，故时间复杂度范围为O（n）~O(n^2)。平均复杂度为O(n^2)。

2.希尔排序，希尔排序是对直接插入排序算法的改进，即选取增量序列做一次直接插入排序，然后缩小增量，直到最后一个增量为1。我上网查了一下，感觉希尔排序的复杂度比较麻烦，没什么参考资料，这里就直接给出答案，O（n）~O(n^2)，我就把它当做直接插入排序记了。ps：好像有人说最好的情况是O（n^1.3），我看到的是平均复杂度为O(n^2),有没有高人出来指点一下！

3.选择排序，选择排序每次选择最大的数据放在数组的后面。每趟都需要对数组进行遍历找出最大的放在后面 ，因此最好的情况一共需要访问1+2+3+…+n-1= n(n-1)/2次，故时间复杂对为O(n^2)。平均时间复杂度为O(n^2)。

4.堆排序，比较好记，堆排序要先构造初始堆，时间复杂度为O（n）,然后排序重建堆的时间复杂度为O（nlog2n）,所以复杂度为O（n）+ O（nlog2n）= O（nlog2n）,原谅楼主非科班出身，对堆啊，二叉树啊的复杂度理解没那么深刻，脑阔疼也不想去推，所以这里就直接记，堆排序的好坏情况都是一样的，时间复杂度为O（nlog2n）。平均时间复杂度O（nlog2n）。

5.冒泡排序，冒泡排序是每次从左边选择一个数据一次与右边数据比较，如果比右边大就交换位置，继续往后比较，那么最好的情况就是序列本身就是升序的，一趟下来比较了n-1次不需要交换结束排序，时间复杂度为O（n）；最坏的情况序列本身降序，那么第一次就需要交换n-1次，第二次排序需要交换n-2次，最后一共需要交换(n-1)+(n-2)+(n-3)+….+1= n(n-1)/2，所以时间复杂度为O（n^2）。故时间复杂度为O（n）~O(n^2)。平均复杂度为O(n^2)。

6.快速排序，参考二楼老哥的说法，快排的原理就是每次从当前数组中选出一个pivot元素，并依此元素遍历数组，将数组划分成两部分:小于pivot的元素和大于pivot的元素。然后在两个子数组中分别递归地进行这个操作。因为是2分，所以一共需要划分log2n次，每次遍历一遍数组，复杂度nlog2n。最坏的情况是每次选出的pivot都是当前数组中最大或最小的元素，此时只能划分出一个子数组(另一个没有元素)。那么一共需要划分n次，每次遍历一遍，时间复杂度为O(n^2)。故时间复杂度为O（nlog2n）~ O(n^2)。平均负载度为O（nlog2n）。

7.归并排序，由于是从序列中间对半分，分治排序，因此不管情况好还都要进行划分log2n次，每次遍历一遍数组，时间复杂度O（nlog2n）。平均复杂度为O（nlog2n）。

8.基数排序，哈哈哈，表示我没看基数排序，后面再补吧。或者谁补充一下

空间复杂度的话，前五个都是O(1),后面快速排序的空间复杂度为O（nlog2n）,归并排序是O（n）,不想推就记一下吧，比好好记。