1. 对一组数据(84,47,25,15,21)排序,数据的排列次序在排序的过程中的变化为(1)84 47 25 15 21 (2)15 47 25 84 21 (3)15 21 25 84 47 (4)15 21 25 47 84 则采用的排序方法是( )选择

2. 下面的排序算法中，初始数据集的排列顺序对算法的性能无影响的是：堆排序

3. 下列排序算法中，在待排序数据有序的情况下，花费时间最多的是（ ）快速排序

4. 使用堆排序方法排序（45，78，57，25，41，89），初始堆为（？）89,78,57,25,41,45

5. 将一个从大到小的数组，用以下排序方法排序成从小到大的，（）最快。堆排序

6. 外排中使用置换选择排序的目的,是为了增加初始归并段的长度()对

7. 当待排序记录已经从小到大排序或者已经从大到小排序时,快速排序的执行时间最省()错

8. 冒泡排序算法的时间复杂度是什么？O(N*N)

9. 由于希尔排序的最后一趟与直接插入排序过程相同,因此前者一定比后者花费的时间更多()错

10. 下列排序方法中，属于稳定排序的是（）归并排序

11. 下列排序方法中，属于不稳定排序的是（）选择排序希尔排序堆排序

12. 在文件"局部有序"或文件长度较小的情况下,最佳内部排序的方法是()直接插入排序

13. 下列序排算法中最坏复杂度不是 n(n-1)/2 的是？堆排序

14. 在一个元素个数为 N 的数组里，找到升序排在 N/5 位置的元素的最优算法时间复杂度是 O(n)

15. 将 N 条长度均为 M 的有序链表进行合并，合并以后的链表也保持有序，时间复杂度为()?O(N * M * logN)

16. 若要求尽可能快地对序列进行稳定的排序,则应选()归并排序

17. 数据序列(8,9,10,4,5,6,20,1,2)只能是下列排序算法中的()的两趟排序后的结果, 插入排序

18. 序列{2,1,4,9,8,10,6,20}是某排序算法第二轮排序的结果，则该算法只能是快速排序

19. 对 n 个元素的数组进行（），其平均时间复杂度和最坏情况下的时间复杂度都是 O（nlogn）. 堆排序

20. 假设线性表的长度为 n，则在最坏情况下，冒泡排序需要的比较次数为多少次？n(n-1)/2

21. 下面的排序算法中，初始数据集的排列顺序对算法的性能无影响的是堆排序

22. 在待排序的元素序列基本有序的前提下，效率最高的排序方法是？插入排序

23. 25，84，21，47，15，27，68，35，20 进行排序时，变化为“20，15，21，25，47，27，68，35，84”“15，20，21，25，35，27，47，68，84”“15，20，21，25，27，35，47，68，84”的排序方法是（）？快速排序

24. 设有 5000 个待排序的记录的关键字，如果需要用最快的方法选出其中最小的 10 个记录关键字，则用下列哪个方法可以达到此目的()堆排序

25. 下述几种排序方法中，要求内存最大的是（）归并排序

26. 将整数数组（7-6-3-5-4-1-2）按照堆排序的方式原地进行升序排列，请问在第一轮排序结束之后，数组的顺序是_____。6-5-3-2-4-1-7

27. 有一个小白程序员，写了一个只能对 5 个数字进行排序的函数。现在有 25 个不重复的数字，请问小白同学最少调几次该函数，可以找出其中最大的三个数？7

28. 线性表的长度为 10，在最坏情况下，冒泡排序需要比较次数为（）。45

29. 使下列算法的时间复杂度描述错误的有？插入排序: O(n*n*n) 归并排序:O(n * n)

30. 对下列四种排序方法,在排序中关键字比较次数同记录初始排列无关的是()折半插入

31. 下面的排序算法中,不稳定的是()简单选择排序, 希尔排序, 堆排序

32. 精俭排序，即一对数字不进行两次和两次以上的比较，以下是“精俭排序”的是插入排序, 归并排序

33. 0~999999 之间的所有数字中，任何一位都不包括数字 3 的数字的总数为____。531441

34. 下面关于排序算法描述正确的是冒泡排序算法平均时间复杂度是 O（N 的平方），平均时间复杂度低的算法不一定是最优算法, 选择排序算法时，需要考虑表中元素的个数

35. 最坏情况下，合并两个大小为 n 的已排序数组所需要的比较次数为 2n-1

36. 设某文件经内排序后得到 100 个初始归并段（初始顺串），若使用多路归并排序算法，且要求三趟归并完成排序，问归并路数最少为 5

37. 外部排序是把外存文件调入内存,可利用内部排序的方法进行排序,因此排序所花的时间取决于内部排序的时间()错

38. 一个有向无环图的拓扑排序序列()是唯一的，不一定

39. 堆排序的时间复杂度是（），堆排序中建堆过程的时间复杂度是（）。 O(n log n),(n)

40. ()占用的额外空间的空间复杂性为 O(1)，堆排序

41. 最坏情况下 insert sort, quick sort ,merge sort 的复杂度分别是多少？O(n*n),O(n*n),O(nlogn)

42. 有些排序算法在每趟排序过程中,都会有一个元素被放置在其最终的位置上,下列算法不会出现此情况的是()shell 排序，在希尔排序中，只有经过最后一趟排序后才会确定每个元素的最终位置

43. 希尔排序法属于哪一种类型的排序法，插入排序

44. 用某种排序方法对关键字序列（25,84,21,47,15,27,68,35,20）进行排序，序列的变化情况采样如下：

20,15,21,25,47,27,68,35,84
15,20,21,25,35,27,47,68,84
15,20,21,25,27,35,47,68,84
请问采用的是以下哪种排序算法（）快排

45. 排序算法中，比较次数与初始序列无关的排序方法有哪些？选择排序

46. 有一组数据(15,9,7,8,20,-1,7,4),用堆排序的筛选方法建立的初始堆为() -1,4,7,8,20,15,7,9

47. 序列{2,1,4,9,8,10,6,20}是某排序算法第二轮排序的结果，则该算法只能是快速排序

48. 在下列排序方法中，不稳定的方法有：堆排序，快排，直接选择排序，希尔排序

49. 在对一组记录(54,38,96,23,15,72,60,45,83)进行直接插入排序时,当把第 7 个记录 60 插入到有序表时,为寻找插入位置需比较()次 3

50. 通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应的位置并插入的排序算法是（）插入排序

51. 在待排序的元素序列基本有序的前提下，效率最高的排序方法是？插入排序或者冒泡排序

52. 已知数据表 A 中每个元素距其最终位置不远，为了节省时间，应该采取的算法是()　直接插入排序

53. 对于基本有序的序列，按照那种排序方式最快：冒泡排序

54. 基本有序反而不好的是：快速排序，基本有序的适合冒泡排序。

55. 不稳定的排序有 4 个：快排，选择排序，堆排序，希尔排序

排序法	平均时间	最坏情况	最好情况	稳定度	额外空间	备注
1.直接插入	$O(n^2)$	$O(n^2)$	$O(n)$	稳定	$O(1)$	大部分已排序时较好（简单）
1．希尔	$O(n\log n)$	$O(n\log n)$	与步长相关	不稳定	$O(1)$	n小时较好（较复杂）
2．冒泡	$O(n^2)$	$O(n^2)$	$O(n)$	稳定	$O(1)$	n小时较好（简单）
2．快排	$O(n\log n)$	$O(n^2)$	$O(n\log n)$	不稳定	$O(\log n)$	n大时较好,基本有序时反而不好（较复杂）
3.直接选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	不稳定	$O(1)$	n小时较好（简单）
3．堆排序	**$O(n\log n)$**	**$O(n\log n)$**	**$O(n\log n)$**	**不稳定**	**$O(1)$**	**n大时较好（较复杂）**
4．归并	**$O(n\log n)$**	**$O(n\log n)$**	**$O(n\log n)$**	**稳定**	**$O(n)$**	**n大时较好（较复杂）**
基数	$O(d(n+r))$	$O(d(n+r))$	$O(d(n+r))$	稳定	$O(r)$	d为位数，r为基数（较复杂）
计数	$O(n+k)$	$O(n+k)$	$O(n+k)$	稳定	$O(n+k)$	优于比较排序法,0~k为数值范围
桶排序	$O(n+c)$	O (nlogn): 所有的元素落到一个桶中	$O(n)$	稳定	$O(n+m)$	n为数的个数，m为桶数 $c = n*(\log n-\log m)$ 桶越多，效率越高，n=m,达到O(n)，但是占用很大的空间，桶内可用快排等

查 · 论 · 编	排序算法
理论	计算复杂性理论 · 大O符号 · 全序关系 · 列表 · 稳定性 · 比较排序 · 自适应排序 · 排序网络 · 整数排序
交换排序	冒泡排序 · 鸡尾酒排序 · 奇偶排序 · 梳排序 · 侏儒排序 · 快速排序 · 臭皮匠排序 · Bogo排序
选择排序	选择排序 · 堆排序 · 平滑排序 · 笛卡尔树排序 · 锦标赛排序 · 圈排序
插入排序	插入排序 · 希尔排序 · 伸展排序 · 二叉查找树排序 · 图书馆排序 · 耐心排序
归并排序	归并排序 · 梯级归并排序 · 振荡归并排序 · 多相归并排序 · 列表排序
分布排序	美国旗帜排序 · 珠排序 · 桶排序 · 爆炸排序 · 计数排序 · 鸽巢排序 · 相邻图排序 · 基数排序 · 闪电排序 · 插值排序
并发排序	双调排序器 · Batcher归并网络 · 两两排序网络
混合排序	区块排序 · Tim排序 · 内省排序 · Spread排序 · J排序
其他	拓扑排序 · 煎饼排序 · 意粉排序

稳定的排序 ［编辑］

- 冒泡排序（bubble sort）— $O(n^2)$
- 插入排序（insertion sort）—$O(n^2)$
- 鸡尾酒排序（cocktail sort）—$O(n^2)$
- 桶排序（bucket sort）—$O(n)$；需要$O(k)$额外空间
- 计数排序（counting sort）—$O(n+k)$；需要$O(n+k)$额外空间
- 归并排序（merge sort）—$O(n \log n)$；需要$O(n)$额外空间
- 原地归并排序— $O(n \log^2 n)$如果使用最佳的现在版本
- 二叉排序树排序（binary tree sort）— $O(n \log n)$期望时间；$O(n^2)$最坏时间；需要$O(n)$额外空间
- 鸽巢排序（pigeonhole sort）—$O(n+k)$；需要$O(k)$额外空间
- 基数排序（radix sort）—$O(n \cdot k)$；需要$O(n)$额外空间
- 侏儒排序（gnome sort）— $O(n^2)$
- 图书馆排序（library sort）— $O(n \log n)$期望时间；$O(n^2)$最坏时间；需要$(1+\varepsilon)n$额外空间
- 块排序（block sort）— $O(n \log n)$

不稳定的排序 ［编辑］

- 选择排序（selection sort）—$O(n^2)$
- 希尔排序（shell sort）—$O(n \log^2 n)$如果使用最佳的现在版本
- Clover排序算法（Clover sort）—O(n)期望时间，$O(n^2)$最坏情况
- 梳排序— $O(n \log n)$
- 堆排序（heap sort）—$O(n \log n)$
- 平滑排序（smooth sort）— $O(n \log n)$
- 快速排序（quick sort）—$O(n \log n)$期望时间，$O(n^2)$最坏情况；对于大的、随机数列表一般相信是最快的已知排序
- 内省排序（introsort）—$O(n \log n)$

名称	数据对象	稳定性	时间复杂度		额外空间复杂度	描述
			平均	最坏		
冒泡排序	数组	✓	$O(n^2)$		$O(1)$	（无序区，有序区）。 从无序区通过交换找出最大元素放到有序区前端。
选择排序	数组	✗	$O(n^2)$		$O(1)$	（有序区，无序区）。 在无序区里找一个最小的元素跟在有序区的后面。对数组：比较得多，换得少。
	链表	✓				
插入排序	数组、链表	✓	$O(n^2)$		$O(1)$	（有序区，无序区）。 把无序区的第一个元素插入到有序区的合适的位置。对数组：比较得少，换得多。
堆排序	数组	✗	$O(n\log n)$		$O(1)$	（最大堆，有序区）。 从堆顶把根卸出来放在有序区之前，再恢复堆。
归并排序	数组	✓	$O(n\log^2 n)$		$O(1)$	把数据分为两段，从两段中逐个选最小的元素移入新数据段的末尾。 可从上到下或从下到上进行。
			$O(n\log n)$		$O(n)+O(\log n)$ 如果不是从下到上	
	链表				$O(1)$	
快速排序	数组	✗	$O(n\log n)$	$O(n^2)$	$O(\log n)$	（小数，基准元素，大数）。 在区间中随机挑选一个元素作基准，将小于基准的元素放在基准之前，大于基准的元素放在基准之后，再分别对小数区与大数区进行排序。
希尔排序	数组	✗	$O(n\log^2 n)$	$O(n^2)$	$O(1)$	每一轮按照事先决定的间隔进行插入排序，间隔会依次缩小，最后一次一定要是1。
计数排序	数组、链表	✓	$O(n+m)$		$O(n+m)$	统计小于等于该元素值的元素的个数i，于是该元素就放在目标数组的索引i位（i≥0）。
桶排序	数组、链表	✓	$O(n)$		$O(m)$	将值为i的元素放入i号桶，最后依次把桶里的元素倒出来。
基数排序	数组、链表	✓	$O(k\times n)$	$O(n^2)$		一种多关键字的排序算法，可用桶排序实现。

- 均按从小到大排列
- k代表数值中的"数位"个数
- n代表数据规模
- m代表数据的最大值减最小值