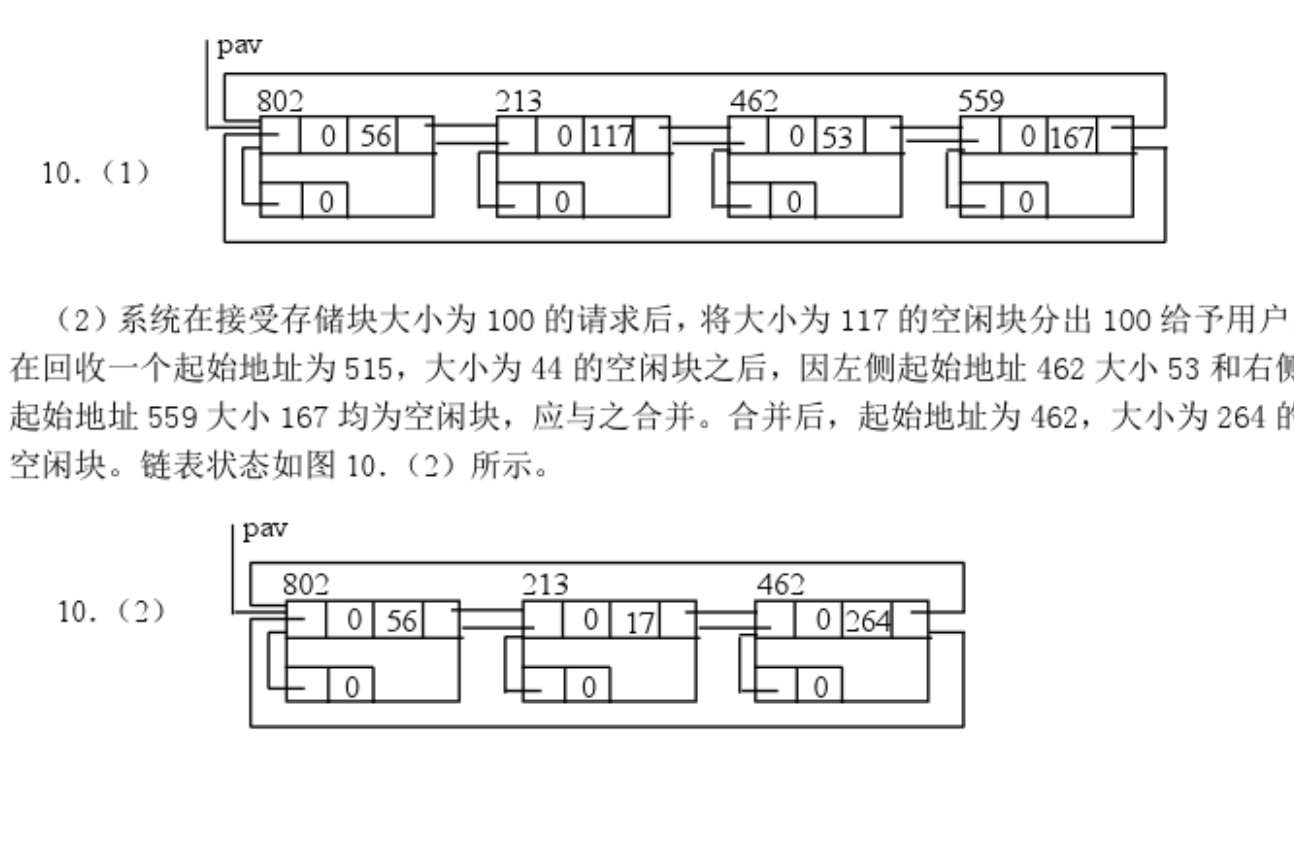
第八、九、十章作业

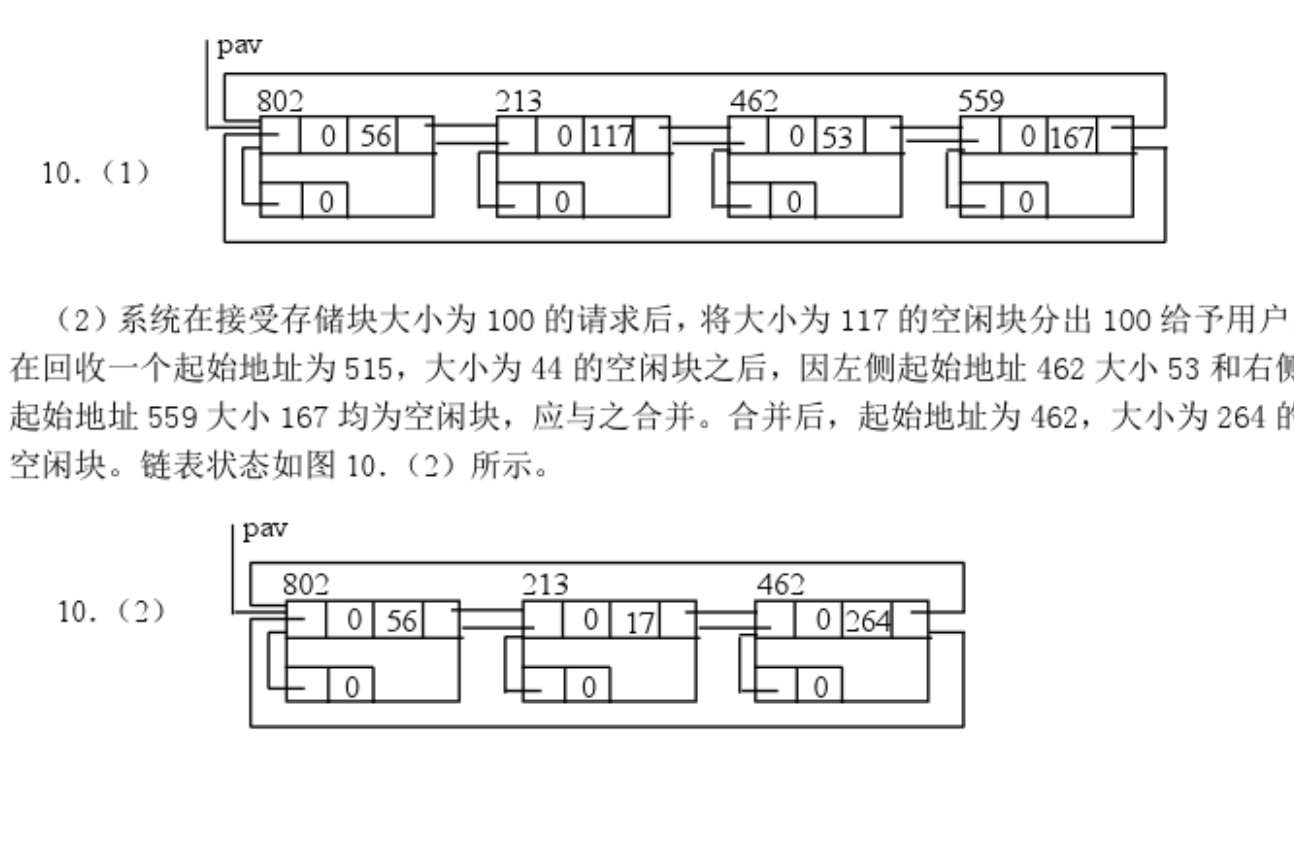
钟赟 2016K8009915009

8.1解：

(1) 系统回收一个起始地址为559，大小为45的空闲块之后，与右侧起始地址604的 空闲块合并为起始地址巍峨559，大小为167的空闲块：

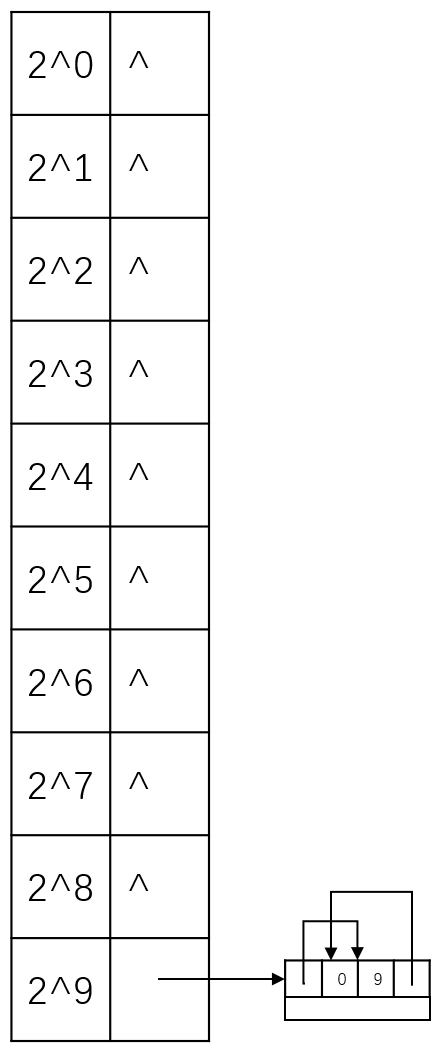


（2）系统首先将起始地址为213大小为117的空闲块分配给100存储块的请求；然后将起始地址为515大小为44的空闲块与它左右的空闲块合并为起始地址为462，大小为264的块：



8.7 解：

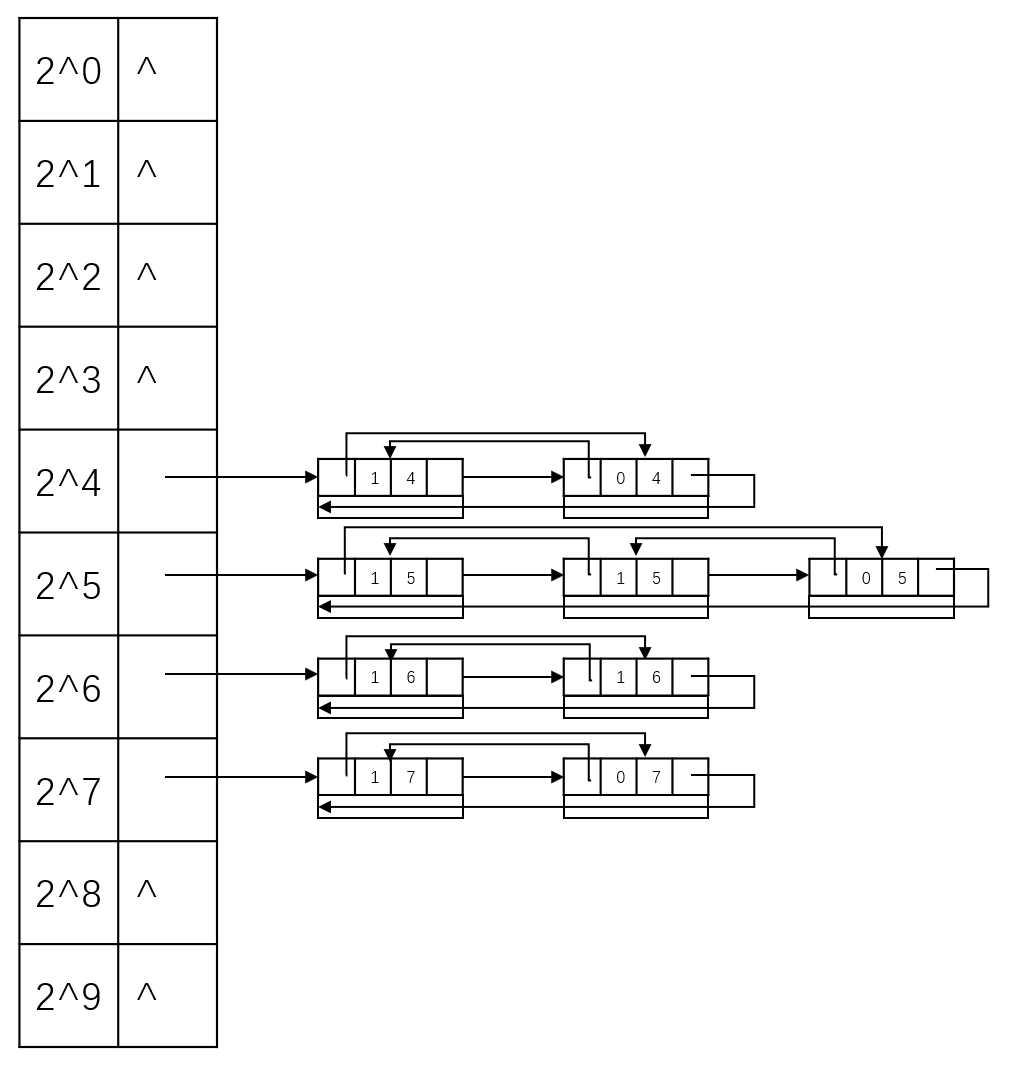
(1) 初始情况：



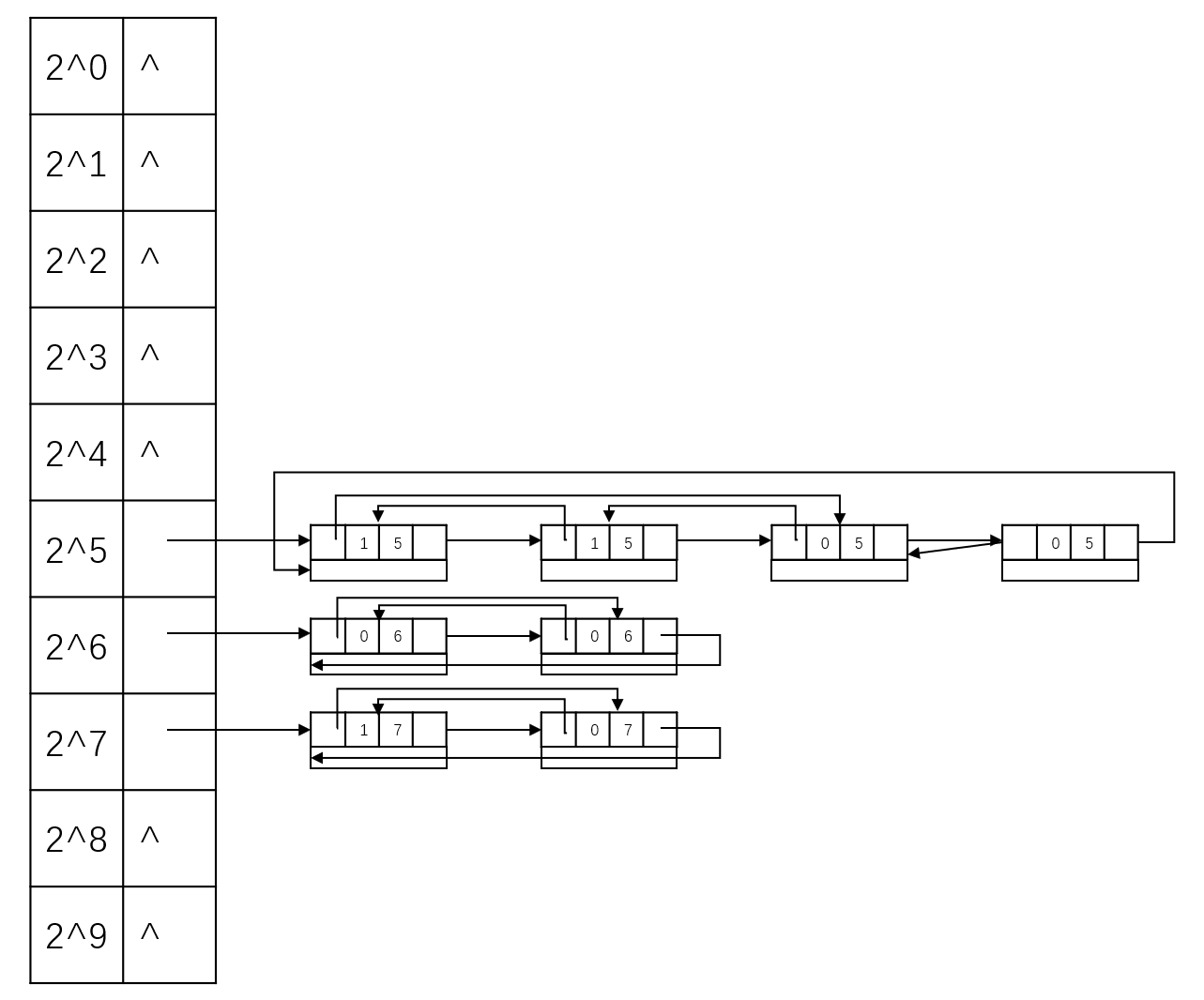
(2) 每个用户所得存储快情况：



六个用户进入后链表状态：



（3）回收后链表状态



9.1 解：

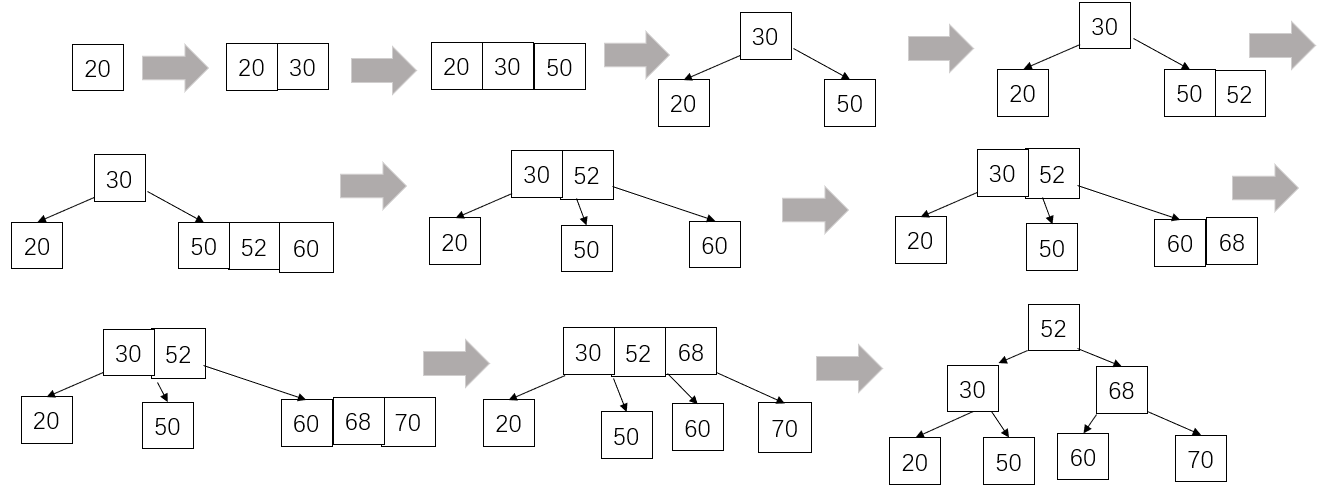
（1）相同。有序顺序表和无序顺序表：n+1

（2）相同。有序顺序表和无序顺序表：(n+1)/2

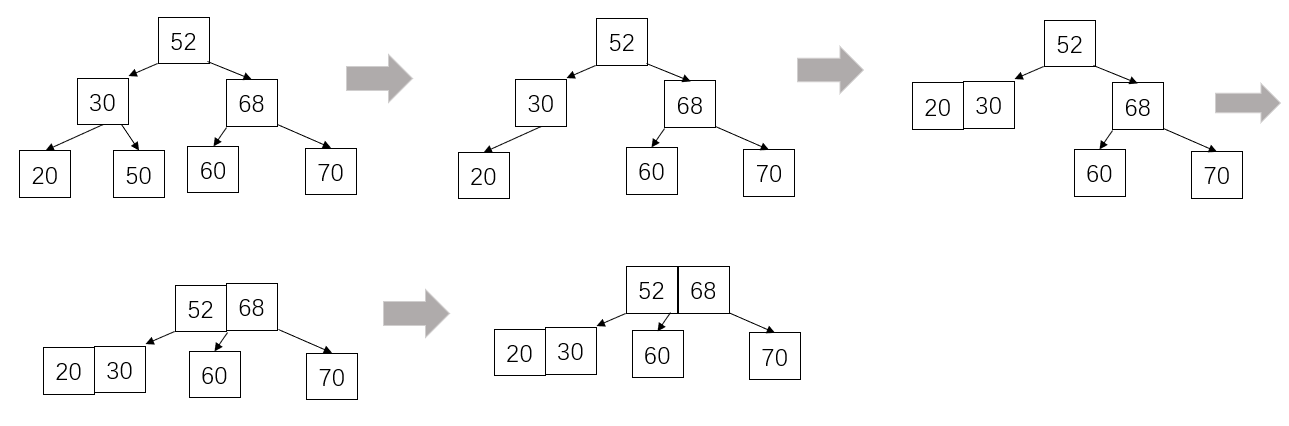
（3）不同。对于有序顺序表，找到第一个与K值相同的关键字后，只需再找到与K不同的元素即可停止查找；对于无序顺序表，需要一直查找到最后一个关键字。

9.14 解：

创建过程：



删除过程：



9.19 解：

H(22) = 0；

H(41) = 2；

H(53) = 5；

H(46) = 6；

H(30) = 2，冲突；H1(30) = 2 + 1\*1 = 3；

H(13) = 6，冲突；H1(13) = 6 + 1\*2 = 8；

H(01) = 3，冲突；H1(01) = 3 + 1\*8 = 0，冲突；H2(01) = 8，冲突；H3(01) = 5，冲突；H4(01) = 2，冲突；H5(01) = 10；

H(67) = 3，冲突；H1(67) = 2，冲突；H2(67) = 1；

哈希表如下：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 哈希地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 关键字 | 22 | 67 | 41 | 30 |  | 53 | 46 |  | 13 |  | 01 |
| 次数 | 1 | 3 | 1 | 2 |  | 1 | 1 |  | 2 |  | 6 |

ASL = (1\*4 + 2\*2 + 3 + 6) / 8 = 17/8

9.24 解：

(1) 在校人数极端情况：，研究生，总共25000人，装载因子α=1

(2) 学号

可以看出聚簇性很好。

(3) α<1，但是无法找到不冲突的H1

10.1 解：

503, 087, 512, 061, 908, 170, 897, 275, 653, 426

直接插入排序：

第一趟：087,503, 512, 061, 908, 170, 897, 275, 653, 426

第二趟：087,503, 512, 061, 908, 170, 897, 275, 653, 426

第三趟：061,087,503, 512, 908, 170, 897, 275, 653, 426

第四趟：061,087,503, 512, 908, 170, 897, 275, 653, 426

第五趟：061,087, 170, 503, 512, 908, 897, 275, 653, 426

第六趟：061,087, 170, 275, 503, 512, 897, 908, 653, 426

第八趟：061,087, 170, 275, 503, 512, 653, 897, 908, 426

第九趟：061,087, 170, 275, 426, 503, 512, 653, 897, 908

希尔排序

(d[1]=5, d[2]=3, d[3]=1)

第一趟：170, 087, 275, 061, 426, 503, 897, 512, 653, 908

第二趟：061, 087, 275, 170, 426, 503, 897, 512, 653, 908

第三趟：061, 087, 170, 275, 426, 503, 512, 653, 897, 908

快速排序

第一趟：(426,087,275,061,170) 503(897,908,653,512)

第二趟：(170, 087,275,061)426, 503(512,653)897(908)

第三趟：(061,087)170(275) 426, 503,512(653) 897, 908

第四趟：061,087, 170, 275, 426, 503, 512, 653, 897, 908

堆排序

建堆：061, 087, 170, 275, 426,512, 897, 503, 653, 908

(输出 061) 087, 275, 170, 503, 426, 512, 897, 653

(输出 087) 170, 275, 512, 503, 426, 653, 897, 908

(输出 170) 275, 406, 512, 503, 908, 653, 897

(输出 275) 406, 503, 512, 897, 908, 653

(输出 406) 503, 653, 512, 897, 908

(输出 503) 512, 653, 908, 897

(输出 512) 653, 897, 908

(输出 653) 897, 908

(输出 897) 908

归并排序

第一趟：（087, 503)(061, 512)(170, 908)(275, 897)(426, 653)

第二趟：（061，087, 503, 512)(170, 275, 897, 908)(426, 653)

第三趟：（061，087, 170, 275, 503, 512, 897, 908)(426, 653)

第四趟：061, 087, 170, 275, 426, 503, 512, 653, 897, 908

基数排序

第一趟：170, 061, 512, 503, 653, 275, 426, 087, 897, 908

第二趟：503, 908, 512, 426, 653, 061, 170, 275, 087, 897

第三趟：061, 087, 170, 275, 426, 503, 512, 653, 897, 908

10.3 解：

（1）直接插入排序 稳定

（2）希尔排序 不稳定

（3）快速排序 不稳定

（4）堆排序 不稳定

（5）归并排序 稳定

（6）基数排序 稳定

10.15 解：

先通过完全二叉树类似进行两两比较，淘汰小的数，直到找出最大的数，需要比较n-1次；

最小的数是被首次淘汰的数，至多有个，从中找出最小的数至多需要次；

所以共需比较次。

10.21 解：

以数列4，3，2，1为例：

折半插入排序

第一趟：4，3，2，1

第二趟：3，4，2，1

比较3，4；

第三趟：2，3，4，1

比较2，3；

第四堂：1，2，3，4

比较1，3；比较1，2；

2-路归并排序法

第一趟：（3，4）（1，2）

比较3，4；比较1，2

第二趟：（1，2，3，4）

比较3，1；比较3，2