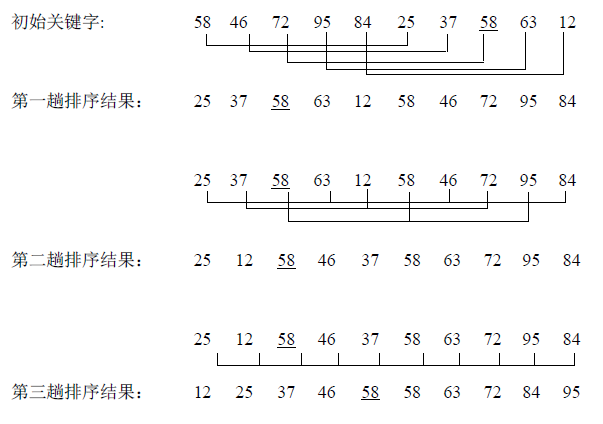
8.2 答：(1) 采用堆排序最好 因为以上几种算法中，快速排序、归并排序和基数排序都是在排序结束后才能确定数据元素的全部顺序，而无法知道排序过程中部分元素的有序性。堆排序则每次输出一个最大(或最小)的元素，然后对堆进行调整，故采用堆排序方法是合适的。

(2) 两个基本操作：比较两个关键字的大小、改变指向记录的指针或移动记录本身。

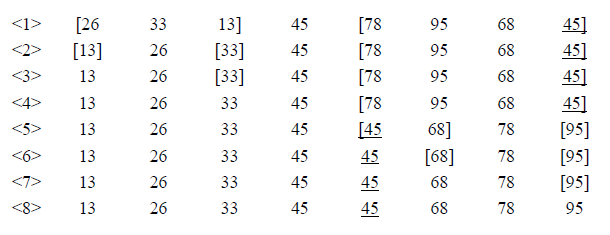
8.5 答：下图给出了希尔排序的整个过程，用同一连线上的关键字表示其所属的记录在同一组。

为区别具有相同关键字 58 的不同记录，用下划线标记后一个记录的关键字。

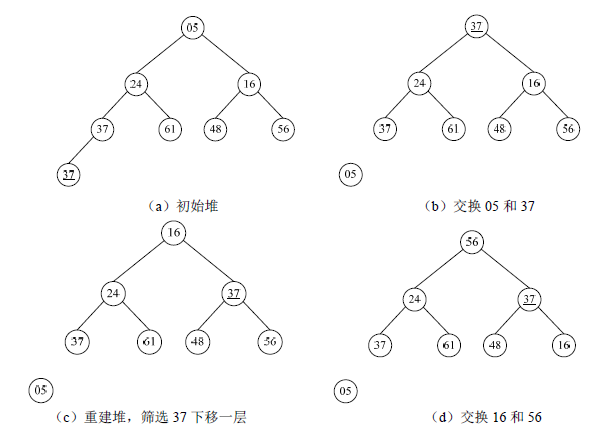


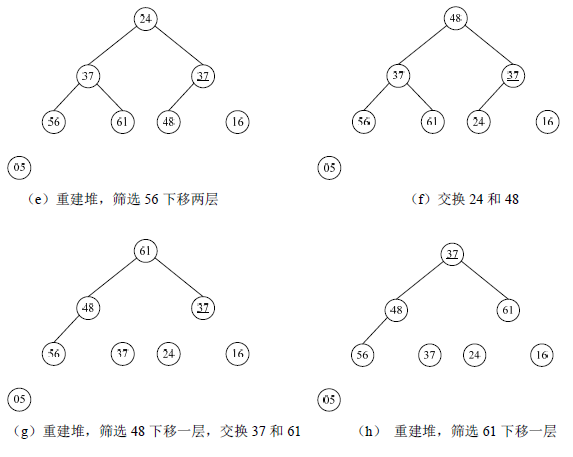
8.6 各趟排序之后的结果如下图所示

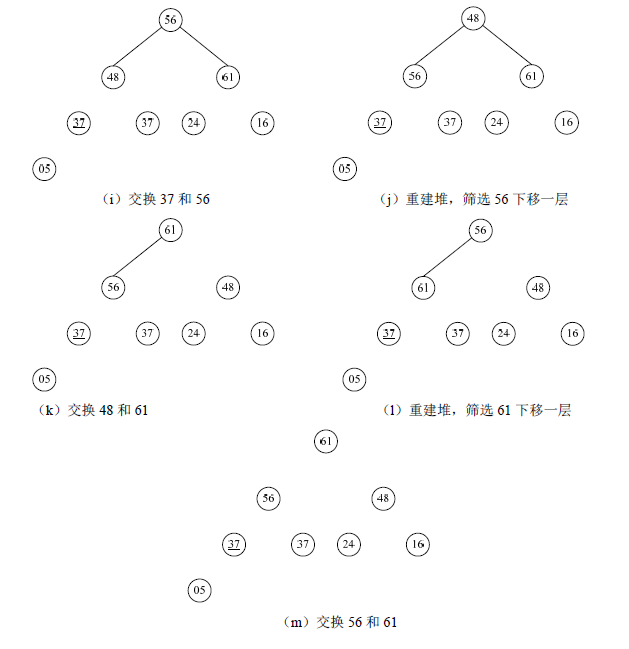
初始关键字序列 [45 33 68 95 78 13 26 45]



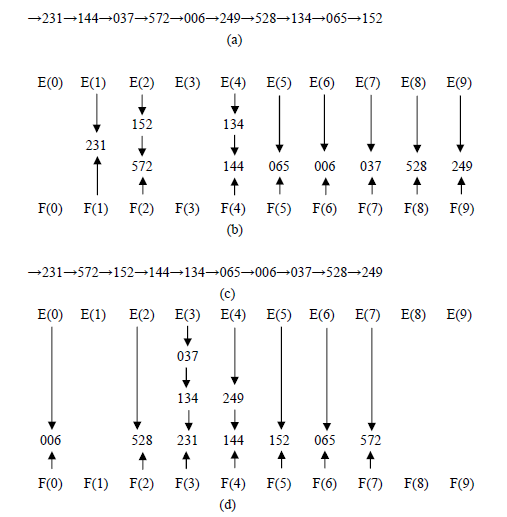
8.7 答：调用筛选运算进行堆排序的过程如 (a)～(m)所示。

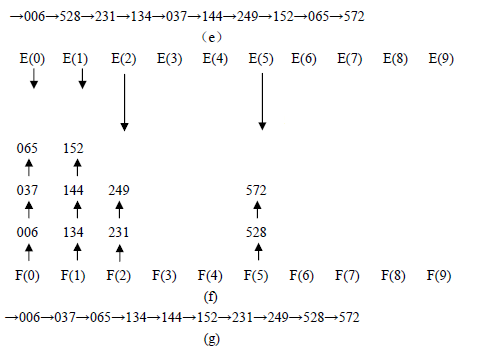






8.9 答：关键字是十进制整数，r=10，d=3，基数排序过程如下图所示。第一趟分配在关键字的个位进行，将链表中的记录分配至 10 个队列中去，每个队列中的记录关键字的个位数相同，如下图(b)所示，其中 F[i]和 E[i]分别为第 i 个队列的头指针和尾指针；第一趟收集是改变所有非空队列的队尾记录的指针域，令其指向下一个非空队列的头指针，重新将 10 个队列中的记录链接成一个链表，如下图(c)所示；第二趟分配，第二趟收集及第三趟分配和第三趟收集分别是对关键字的十位数和百位数进行的，其过程和个位数相同，如下图(d)至(g)所示。至此排序完毕。





8.12 答：实现本题功能的算法如下：

void SelectSort (RecType R[],int n)

{

int i,k,j;

for (i=1;j<n;i++)

{

k=i;

for(j = i+1;j<n;j++)

if(R[j].score>R[k].score)

k=j;

if(i != k )

{

R[0] = R[i];

R[i] = R[k];

R[k] = R[0];

}

}

for(i=1;i<n;i++)

{

printf(“%d”,R[i].num);

printf(“%3.1f”,R[i].score);

if (i%10==0)

printf(“\n”);

}