1. 若对 27 个元素只进行 3 趟多路归并排序,则选取的归并路数为_____。

A、2　B、3　C、4　D、5

答案：B

解答：设需要的路数位 m，则对于 n 个元素做 m 路归并排序所需趟数 $s=\log_m n$。当 n=27，s=3 时，则 m=3。

2. 如果将所有中国人按照生日(不考虑年份，只考虑月、日)来排序，那么使用_____排序算法最快。

A、归并排序　B、希尔排序　C、快速排序　D、基数排序

答案：D

解答：按照所有中国人的生日（月、日）排序，一方面是 n 很大；另一方面 d 不大（d=2，两个排序码）且一个排序码的基数为 12（月），另一排序码的基数为 31（日），都不大，可采用多排序码排序，即基数排序，其时间复杂度可达 O(n)。

3. 下列排序算法中，_____算法是不稳定的。

A、起泡排序　B、直接插入排序　C、基数排序　D、快速排序

答案：D

解答：不稳定的排序算法有希尔排序、简单选择排序、堆排序和快速排序。

4. 具有 n 个关键字的 m 阶 B 树有_____个失败结点。

答案：n+1

解答：m 阶 B 树的失败结点即为查找失败走到的结点，对 n 个关键字查找不成功的情况是待查找的关键字值介于 n 个关键字中某两个关键字值之间，这样的可能性有 n+1 种。

5. 假定有 k 个关键字互为同义词，若用线性探测法把这 k 个关键字值存入散列表肿，至少要进行_____次探测。

答案：(k+1)×k/2

解答：探查次数最少的情况是第 1 个关键字通过 1 次比较后插入，第 2 个关键字通过 2 次比较后插入……第 k 各关键字则通过 k 次比较后插入。因此，总的探查次数=1+2+…+k。

6. 根据初始关键字序列(19，22，01，38，10)建立的二叉排序树的高度为_____。

答案：3

7. 设一组初始记录关键字序列为(45，80，48，40，22，78)，则分别给出第 4 趟简单选择排序和第 4 趟直接插入排序后的结果。

答案：(22，40，45，48，80，78)，(40，45，48，80，22，78)

解答：请参考课本或 PPT 的示意图。

8. 已知待散列的线性表为（36，15，40，63，22），散列用的一维地址空间为[0..6]，假定选用的散列函数是 H（K）= K mod 7，若发生冲突采用线性探查法处理，试：

（1）计算出每一个元素的散列地址后，在下图中填写出散列表：

0	1	2	3	4	5	6

（2）求出在查找每一个元素概率相等情况下的平均查找长度。

答案：

(1)

0	1	2	3	4	5	6
63	36	15	22		40	

(2)　　ASL= $\frac{1+2+1+1+3}{5}=1.6$

解答： H(36)=36 mod 7=1;　　H_1(22)=(1+1) mod 7=2; ….冲突

H(15)=15 mod 7=1;….冲突　　H_2(22)=(2+1) mod 7=3;

H_1 (15)=(1+1) mod 7=2;
H(40)=40 mod 7=5;
H(63)=63 mod 7=0;

H(22)=22 mod 7=1; ….冲突

9. 设计在链式结构上实现简单选择排序算法，请在下划线处填上正确的语句。

```
void simpleselectsorlklist(lklist *&head)
{
  lklist *p,*q,*s;   int min,t;
  if(head==0 ||head->next==0) return;
  for(q=head; q!=0;q=q->next)
  {
    min=q->data; s=q;
    for(p=q->next; p!=0;p=p->next)
        ____________________________
    if(s!=q)
        {t=s->data; s->data=q->data; q->data=t;}
  }
}
```

答案： **if(min>p->data){min=p->data; s=p;}**

10. 一个长度为 L（L>=1）的长序序列 S，处在第⌊L/2⌋个位置的数称为 S 的中位数。例如，若序列 S1=（11，13，15，17，19），则 S1 的中位数是 15，两个序列的中位数是含它们所有元素的升序序列的中位数。例如，若 S2=（2，4，6，8，20），则 S1 和 S2 的中位数是 11。现在有两个等长升序序列 A 和 B，试设计一个在时间和空间两方面都尽可能高效的算法，找出两个序列 A 和 B 的中位数。

要求：

（1）给出算法的基本设计思想。

（2）根据设计思想，采用 C++语言描述算法。

（3）说明你所设计算法的时间复杂度和空间复杂度。

答案：（1）算法的基本设计思想如下：

分别求出序列 A 和 B 的中位数，设为 a 和 b，求序列 A 和 B 的中位数过程如下：

①若 a=b，则 a 或 b 即为所求中位数，算法结束；

②若 a<b，则舍弃序列 A 中较小的一半，同时舍弃序列 B 中较大的一半，要求舍弃的长度相等；

③若 a>b，则舍弃序列 A 中较大的一半，同时舍弃序列 B 中较小的一半，要求舍弃的长度相等；

在保留的两个长序序列中，重复过程①②③，直到两个序列中只含有一个元素时为止，较小者即为所求的中位数。

（2）算法的描述如下：

```
int midvalue(int a[], int b[], int L)
{
    s1=s2=0; e1=e2=L-1;
    while(s1<e1)
    {
        len=e1-s1+1;
        m1=s1+(e1-s1)/2;
        m2=s2+(e2-s2)/2;
        if (a[m1]==b[m2])
            return a[m1];
        else if (a[m1]< b[m2])
            if (len%2)
                { s1=m1; e2=m2;}
            else { s1=m1+1; e2=m2-1; }
```

```
            else
          if (len%2)
              { s2=m2; e1=m1;}
          else { s2=m2+1; e1=m1-1; }
      } //end while
    return a[s1];
}//end midvalue()
```

（3）算法的时间复杂度为 O(L)和空间复杂度为 O(1)