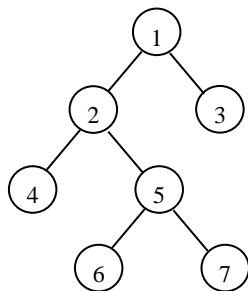


2009-2016 全国硕士生联考 数据结构部分试题及参考答案

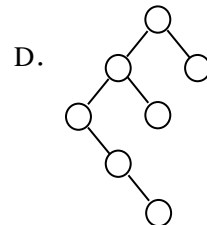
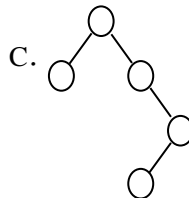
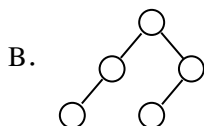
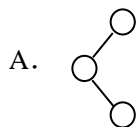
2009 年试卷

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

- 为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是
A. 栈 B. 队列 C. 树 D. 图
- 设栈 S 和队列 Q 的初始状态均为空，元素 a, b, c, d, e, f, g 依次进入栈 S。若每个元素出栈后立即进入队列 Q，且 7 个元素出队的顺序是 b, d, c, f, e, a, g，则栈 S 的容量至少是
A. 1 B. 2 C. 3 D. 4
- 给定二叉树如下图所示。设 N 代表二叉树的根，L 代表根结点的左子树，R 代表根结点的右子树。若遍历后的结点序列为 3, 1, 7, 5, 6, 2, 4，则其遍历方式是



- A. LRN B. NRL C. RLN D. RNL
- 下列二叉排序树中，满足平衡二叉树定义的是



- 已知一棵完全二叉树的第 6 层（设根为第 1 层）有 8 个叶结点，则该完全二叉树的结点个数最多是

- A. 39 B. 52 C. 111 D. 119
6. 将森林转换为对应的二叉树，若在二叉树中，结点 u 是结点 v 的父结点的父结点，则在原来的森林中， u 和 v 可能具有的关系是
- I. 父子关系 II. 兄弟关系 III. u 的父结点与 v 的父结点是兄弟关系
- A. 只有 II B. I 和 II C. I 和 III D. I、II 和 III
7. 下列关于无向连通图特性的叙述中，正确的是
- I. 所有顶点的度之和为偶数 II. 边数大于顶点个数减 1
- III. 至少有一个顶点的度为 1
- A. 只有 I B. 只有 II C. I 和 II D. I 和 III
8. 下列叙述中，不符合 m 阶 B 树定义要求的是
- A. 根结点最多有 m 棵子树 B. 所有叶结点都在同一层上
- C. 各结点内关键字均升序或降序排列 D. 叶结点之间通过指针链接
9. 已知关键字序列 5, 8, 12, 19, 28, 20, 15, 22 是小根堆（最小堆），插入关键字 3，调整后得到的小根堆是
- A. 3, 5, 12, 8, 28, 20, 15, 22, 19 B. 3, 5, 12, 19, 20, 15, 22, 8, 28
- C. 3, 8, 12, 5, 20, 15, 22, 28, 19 D. 3, 12, 5, 8, 28, 20, 15, 22, 19
10. 若数据元素序列 11, 12, 13, 7, 8, 9, 23, 4, 5 是采用下列排序方法之一得到的第二趟排序后的结果，则该排序算法只能是
- A. 起泡排序 B. 插入排序 C. 选择排序 D. 二路归并排序

二、综合应用题：41~47 小题，共 70 分。请将答案写在答题纸指定位置上。

41. (10 分) 带权图（权值非负，表示边连接的两顶点间的距离）的最短路径问题是找出从初始顶点到目标顶点之间的一条最短路径。假设从初始顶点到目标顶点之间存在路径，现有一种解决该问题的方法：

- ① 设最短路径初始时仅包含初始顶点，令当前顶点 u 为初始顶点；
- ② 选择离 u 最近且尚未在最短路径中的一个顶点 v ，加入到最短路径中，修改当前顶点 $u = v$ ；
- ③ 重复步骤②，直到 u 是目标顶点时为止。

请问上述方法能否求得最短路径？若该方法可行，请证明之；否则，请举例说明。

42. (15 分) 已知一个带有表头结点的单链表，结点结构为

data	link
------	------

，假设该链表

只给出了头指针 list。在不改变链表的前提下, 请设计一个尽可能高效的算法, 查找链表中倒数第 k 个位置上的结点 (k 为正整数)。若查找成功, 算法输出该结点的 data 域的值, 并返回 1; 否则, 只返回 0。要求:

- (1) 描述算法的基本设计思想;
- (2) 描述算法的详细实现步骤;
- (3) 根据设计思想和实现步骤, 采用程序设计语言描述算法(使用 C 或 C++ 或 JAVA 语言实现), 关键之处请给出简要注释。

2009 年答案

一、单项选择题: 每小题 2 分, 共 80 分。

- | | | | | | |
|------|------|------|-------|------|------|
| 1. B | 2. C | 3. D | 4. B | 5. C | 6. B |
| 7. A | 8. D | 9. A | 10. B | | |

二、综合应用题: 共 70 分。

41. 答案要点:

该方法不一定能(或不能)求得最短路径。(4 分)

举例说明:(6 分)

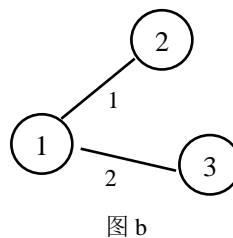
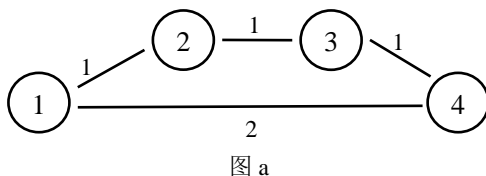


图 a 中, 设初始顶点为 1, 目标顶点为 4, 欲求从顶点 1 到顶点 4 之间的最短路径。显然这两点之间的最短路径长度为 2。但利用给定方法求得的路径长度为 3, 这条路径并不是这两点之间的最短路径。

图 b 中, 设初始顶点为 1, 目标顶点为 3, 欲求从顶点 1 到顶点 3 之间的最短路径。利用给定的方法, 无法求出顶点 1 到顶点 3 的路径。

【评分说明】

①若考生回答“能求得最短路径”, 无论给出何种证明, 均不给分。

②考生只要举出类似上述的一个反例说明“不能求得最短路径”或答案中体现了“局部最优不等于全局最优”的思想, 均可给 6 分; 若举例说明不完全正确, 可酌情给分。

42. 答案要点:

- (1) 算法的基本设计思想:(5 分)

定义两个指针变量 p 和 q ，初始时均指向头结点的下一个结点。 p 指针沿链表移动；当 p 指针移动到第 k 个结点时， q 指针开始与 p 指针同步移动；当 p 指针移动到链表最后一个结点时， q 指针所指元素为倒数第 k 个结点。

以上过程对链表仅进行一遍扫描。

(2) 算法的详细实现步骤：(5 分)

- ① $count = 0$ ， p 和 q 指向链表表头结点的下一个结点；
- ② 若 p 为空，转⑤；
- ③ 若 $count$ 等于 k ，则 q 指向下一个结点；否则， $count = count + 1$ ；
- ④ p 指向下一个结点，转步骤②；
- ⑤ 若 $count$ 等于 k ，则查找成功，输出该结点的 $data$ 域的值，返回 1；否则，查找失败，返回 0；
- ⑥ 算法结束。

(3) 算法实现：(5 分)

```
typedef struct LNode {
    int data;
    struct LNode *link;
} * LinkList;

int SearchN (LinkList list, int k)
{
    LinkList p, q;
    int count = 0;                /* 计数器赋初值 */
    p = q = list->link;          /* p 和 q 指向链表表头结点的下一个结点 */
    while (p != NULL)
    {
        if (count < k)
            count++;              /* 计数器+1 */
        else
        {
            q = q->link;          /* q 移到下一个结点 */
            p = p->link;          /* p 移到下一个结点 */
        }
    }
}
```

```

if (count < k)                                /* 如果链表的长度小于 k */
    return (0);                                /* 查找失败 */
else
{
    printf("%d", q->data);                      /* 查找成功 */
    return (1);
}
}

```

【评分说明】

①若所给出的算法采用一遍扫描方式就能得到正确结果，可给满分 15 分；若采用两遍或多遍扫描才能得到正确结果的，最高给 10 分。若采用递归算法得到正确结果的，最高给 10 分；若实现的算法的空间复杂度过高（使用了大小与 k 有关的辅助数组），但结果正确，最高给 10 分。

②参考答案中只给出了使用 C 语言的版本，使用 C++/JAVA 语言正确实现的算法同样给分。

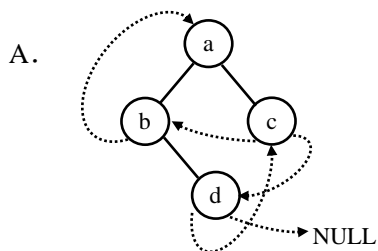
③若在算法基本思想描述和算法步骤描述中因文字表达没有非常清晰地反映出算法的思路，但在算法实现中能够清晰看出算法思想和步骤且正确，按照①的标准给分。

④若考生的答案中算法基本思想描述、算法步骤描述或算法实现中部分正确，可酌情给分。

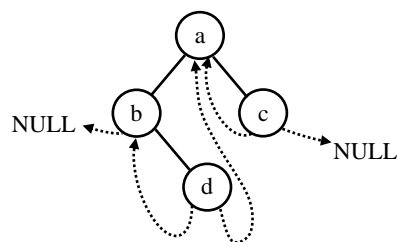
2010 年试卷

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

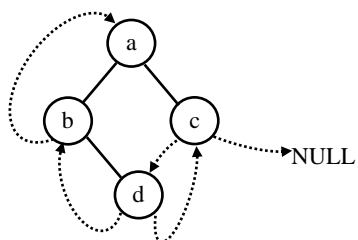
- 若元素 a, b, c, d, e, f 依次进栈，允许进栈、退栈操作交替进行，但不允许连续三次进行退栈操作，则不可能得到的出栈序列是
A. d, c, e, b, f, a B. c, b, d, a, e, f C. b, c, a, e, f, d D. a, f, e, d, c, b
- 某队列允许在其两端进行入队操作，但仅允许在一端进行出队操作。若元素 a, b, c, d, e 依次入此队列后再进行出队操作，则不可能得到的出队序列是
A. b, a, c, d, e B. d, b, a, c, e C. d, b, c, a, e D. e, c, b, a, d
- 下列线索二叉树中（用虚线表示线索），符合后序线索树定义的是



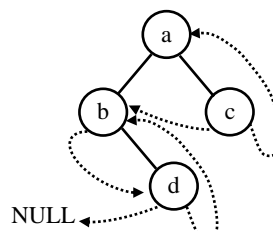
B.
5 / 40



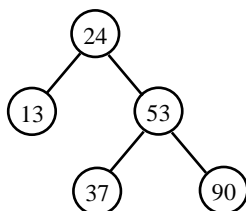
C.



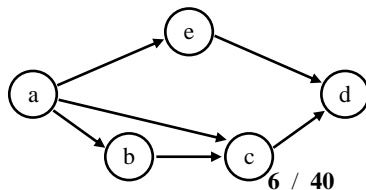
D.



4. 在下图所示的平衡二叉树中，插入关键字 48 后得到一棵新平衡二叉树。在新平衡二叉树中，关键字 37 所在结点的左、右子结点中保存的关键字分别是



- A. 13、48 B. 24、48 C. 24、53 D. 24、90
5. 在一棵度为 4 的树 T 中，若有 20 个度为 4 的结点，10 个度为 3 的结点，1 个度为 2 的结点，10 个度为 1 的结点，则树 T 的叶结点个数是
- A. 41 B. 82 C. 113 D. 122
6. 对 n ($n \geq 2$) 个权值均不相同的字符构造哈夫曼树。下列关于该哈夫曼树的叙述中，错误的是
- A. 该树一定是一棵完全二叉树
- B. 树中一定没有度为 1 的结点
- C. 树中两个权值最小的结点一定是兄弟结点
- D. 树中任一非叶结点的权值一定不小于下一层任一结点的权值
7. 若无向图 $G = (V, E)$ 中含有 7 个顶点，要保证图 G 在任何情况下都是连通的，则需要的边数最少是
- A. 6 B. 15 C. 16 D. 21
8. 对下图进行拓扑排序，可以得到不同拓扑序列的个数是



- A. 4 B. 3 C. 2 D. 1
9. 已知一个长度为 16 的顺序表 L ，其元素按关键字有序排列。若采用折半查找法查找一个 L 中不存在的元素，则关键字的比较次数最多是
- A. 4 B. 5 C. 6 D. 7
10. 采用递归方式对顺序表进行快速排序。下列关于递归次数的叙述中，正确的是
- A. 递归次数与初始数据的排列次序无关
- B. 每次划分后，先处理较长的分区可以减少递归次数
- C. 每次划分后，先处理较短的分区可以减少递归次数
- D. 递归次数与每次划分后得到的分区的处理顺序无关
11. 对一组数据 (2, 12, 16, 88, 5, 10) 进行排序，若前三趟排序结果如下：
- 第一趟排序结果：2, 12, 16, 5, 10, 88
- 第二趟排序结果：2, 12, 5, 10, 16, 88
- 第三趟排序结果：2, 5, 10, 12, 16, 88
- 则采用的排序方法可能是
- A. 起泡排序 B. 希尔排序 C. 归并排序 D. 基数排序

二、综合应用题：41~47 小题，共 70 分。请将答案写在答题纸指定位置上。

41. (10 分) 将关键字序列 (7, 8, 30, 11, 18, 9, 14) 散列存储到散列表中，散列表的存储空间是一个下标从 0 开始的一维数组，散列函数为： $H(\text{key}) = (\text{key} \times 3) \text{ MOD } 7$ ，处理冲突采用线性探测再散列法，要求装填（载）因子为 0.7。
- (1) 请画出所构造的散列表。
- (2) 分别计算等概率情况下查找成功和查找不成功的平均查找长度。
42. (13 分) 设将 n ($n > 1$) 个整数存放于一维数组 R 中。试设计一个在时间和空间两方面都尽可能高效的算法，将 R 中保存的序列循环左移 p ($0 < p < n$) 个位置，即将 R 中的数据由 $(x_0, x_1, \dots, x_{n-1})$ 变换为 $(x_p, x_{p+1}, \dots, x_{n-1}, x_0, x_1, \dots, x_{p-1})$ 。要求：
- (1) 给出算法的基本设计思想。
- (2) 根据设计思想，采用 C 或 C++ 或 JAVA 语言描述算法，关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

2010 年答案

一、单项选择题：每小题 2 分，共 80 分。

1. D 2. C 3. D 4. C 5. B

6. A 7. C 8. B 9. B 10. D
11. A

二、综合应用题：41~47 小题，共 70 分。

41. 答案要点：

(1) 计算的中间结果如下表所示：

关键字	7	8	30	11	18	9	14
散列地址	0	3	6	5	5	6	0
探查次数	1	1	1	1	3	3	2

构造的散列表如下：(6 分)

下标	0	1	2	3	4	5	6	7	8	9
关键字	7	14		8		11	30	18	9	

【评分说明】所画的散列表长度正确（长度为 10，下标从 0 到 9），给 1 分。插入的关键字中，前 4 个无冲突的关键字填写正确，给 2 分。后 3 个关键字填写正确，每个给 1 分。考生可以不写出计算过程。若解答部分正确，酌情给分。

(2) 查找成功的平均查找长度：(2 分)

因为查找长度为 1 的关键字有 4 个，查找长度为 2 的关键字有 1 个，查找长度为 3 的关键字有 2 个，所以查找成功的平均查找长度： $ASL_{成功} = (1 \text{ 次} \times 4 + 2 \text{ 次} \times 1 + 3 \text{ 次} \times 2) / 7 = 12/7$ 。

查找不成功的平均查找长度：(2 分)

各位置对应的探查次数如下表所示：

下标	0	1	2	3	4	5	6
探查次数	3	2	1	2	1	5	4

所以查找不成功的平均查找长度： $ASL_{不成功} = (3+2+1+2+1+5+4)/7 = 18/7$ 。

【评分说明】若分别采用公式（ α 为装填因子）：

$$\frac{1}{2}(1+\frac{1}{1-\alpha}) \text{ 和 } \frac{1}{2}(1+\frac{1}{(1-\alpha)^2})$$

计算查找成功和查找不成功的平均查找长度，且计算正确，可共给 2 分。若解答部分正确，酌情给分。

42. 答案要点：

(1) 给出算法的基本设计思想：(4 分)

先将 n 个数据 $x_0, x_1, \dots, x_p, \dots, x_{n-1}$ 原地逆置，得到 $x_{n-1}, \dots, x_p, x_{p-1}, \dots, x_0$ ，然后再将前 $n-p$ 个和后 p 个元素分别原地逆置，得到最终结果： $x_p, x_{p+1}, \dots, x_{n-1}, x_0, x_1, \dots, x_{p-1}$ 。

算法可以用两个函数实现。一个是逆置函数 `reverse()`，它将给定的数据逆置。另一个是循环左移函数 `leftShift()`，它调用 `reverse()` 函数三次，实现相应功能。

(2) 算法实现：(7 分)


```

void reverse(int r[], int left, int right)
{
    int k = left, j = right, temp;        // k 等于左边界 left, j 等于右边界 right

    while(k < j)
    {
        //交换 r[k]与 r[j]
        temp = r[k];
        r[k] = r[j];
        r[j] = temp;
        k++;                            // k 右移一个位置
        j--;                            // j 左移一个位置
    }
}

void leftShift(int r[], int n, int p)
{
    if (p > 0 && p < n)
    {
        reverse( r, 0, n-1 );          // 将全部数据逆置
        reverse( r, 0, n-p-1 );        // 将前 n-p 个元素逆置
        reverse( r, n-p, n-1 );        // 将后 p 个元素逆置
    }
}

```

【(1)(2) 的评分说明】

①若考生所给算法的时间复杂度及空间复杂度分别为 $O(n)$ 和 $O(1)$ ，且算法实现正确，可给 11 分；若时间复杂度超过 $O(n)$ 或空间复杂度超过 $O(1)$ ，能得到正确结果的，则最高可给 7 分；若时间复杂度超过 $O(n)$ 且空间复杂度超过 $O(1)$ ，能得到正确结果的，则最高可给 3 分。

②若在算法的基本设计思想描述中因文字表达没有非常清晰反映出算法思路，但在算法实现中能够清晰看出算法思想且正确的，可参照①的标准给分。

③若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。

④参考答案中只给出了使用 C 语言的版本，使用 C++/JAVA 语言的答案视同使用 C 语言。

(3) 说明算法复杂性：(2 分)

参考答案给出的算法的时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。

【评分说明】若考生所估计的时间复杂度及空间复杂度与考生所实现的算法一致，可给 2 分。

2011 年试卷

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。

1. 设 n 是描述问题规模的非负整数，下面程序片段的时间复杂度是
 $x = 2;$
 while ($x < n/2$)
 $x = 2 * x;$
 A. $O(\log_2 n)$ B. $O(n)$ C. $O(n \log_2 n)$ D. $O(n^2)$
2. 元素 a, b, c, d, e 依次进入初始为空的栈中，若元素进栈后可停留、可出栈，直到所有元素都出栈，则在所有可能的出栈序列中，以元素 d 开头的序列个数是
 A. 3 B. 4 C. 5 D. 6
3. 已知循环队列存储在一维数组 $A[0..n-1]$ 中，且队列非空时 front 和 rear 分别指向队头元素和队尾元素。若初始时队列为空，且要求第 1 个进入队列的元素存储在 $A[0]$ 处，则初始时 front 和 rear 的值分别是
 A. 0, 0 B. 0, $n-1$ C. $n-1$, 0 D. $n-1$, $n-1$
4. 若一棵完全二叉树有 768 个结点，则该二叉树中叶结点的个数是
 A. 257 B. 258 C. 384 D. 385
5. 若一棵二叉树的前序遍历序列和后序遍历序列分别为 1, 2, 3, 4 和 4, 3, 2, 1，则该二叉树的中序遍历序列不会是
 A. 1, 2, 3, 4 B. 2, 3, 4, 1 C. 3, 2, 4, 1 D. 4, 3, 2, 1
6. 已知一棵有 2011 个结点的树，其叶结点个数为 116，该树对应的二叉树中无右孩子的结点个数是
 A. 115 B. 116 C. 1895 D. 1896
7. 对于下列关键字序列，不可能构成某二叉排序树中一条查找路径的序列是
 A. 95, 22, 91, 24, 94, 71 B. 92, 20, 91, 34, 88, 35
 C. 21, 89, 77, 29, 36, 38 D. 12, 25, 71, 68, 33, 34
8. 下列关于图的叙述中，正确的是
 I. 回路是简单路径
 II. 存储稀疏图，用邻接矩阵比邻接表更省空间
 III. 若有向图中存在拓扑序列，则该图不存在回路
 A. 仅 II B. 仅 I、II C. 仅 III D. 仅 I、III
9. 为提高散列（Hash）表的查找效率，可以采取的正确措施是
 I. 增大装填（载）因子
 II. 设计冲突（碰撞）少的散列函数

III. 处理冲突（碰撞）时避免产生聚集（堆积）现象

- A. 仅 I B. 仅 II C. 仅 I、II D. 仅 II、III

10. 为实现快速排序算法，待排序序列宜采用的存储方式是

- A. 顺序存储 B. 散列存储 C. 链式存储 D. 索引存储

11. 已知序列 25, 13, 10, 12, 9 是大根堆，在序列尾部插入新元素 18，将其再调整为大根堆，调整过程中元素之间进行的比较次数是

- A. 1 B. 2 C. 4 D. 5

二、综合应用题：41~47 小题，共 70 分。

41. （8 分）已知有 6 个顶点（顶点编号为 0~5）的有向带权图 G，其邻接矩阵 A 为上三角矩阵，按行为主序（行优先）保存在如下的一维数组中。

4	6	∞	∞	∞	5	∞	∞	∞	4	3	∞	∞	3	3
---	---	----------	----------	----------	---	----------	----------	----------	---	---	----------	----------	---	---

要求：

- （1）写出图 G 的邻接矩阵 A。
- （2）画出有向带权图 G。
- （3）求图 G 的关键路径，并计算该关键路径的长度。

42. （15 分）一个长度为 L ($L \geq 1$) 的升序序列 S，处在第 $\lceil L/2 \rceil$ 个位置的数称为 S 的中位数。例如，若序列 $S_1=(11, 13, 15, 17, 19)$ ，则 S_1 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 $S_2=(2, 4, 6, 8, 20)$ ，则 S_1 和 S_2 的中位数是 11。现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 或 C++ 或 JAVA 语言描述算法，关键之处给出注释。
- （3）说明你所设计算法的时间复杂度和空间复杂度。

2011 年答案

一、单项选择题：每小题 2 分，共 80 分。

1. A 2. B 3. B 4. C 5. C
6. D 7. A 8. C 9. D 10. A
11. B

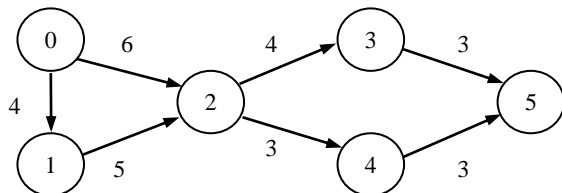
二、综合应用题：41~47 小题，共 70 分。

41. 【答案要点】

- （1）图 G 的邻接矩阵 A 如下：（2 分）

0	4	6	∞	∞	∞
∞	0	5	∞	∞	∞
∞	∞	0	4	3	∞
∞	∞	∞	0	∞	3
∞	∞	∞	∞	0	3
∞	∞	∞	∞	∞	0

(2) 图 G 如下: (2 分)



【评分说明】

- ①若考生画出的图 G 与上图同构, 给 2 分。
- ②若考生画出的图 G 部分正确, 可酌情给分。

(3) 下图中双线箭头所标识的 4 个活动组成图 G 的关键路径。(3 分)

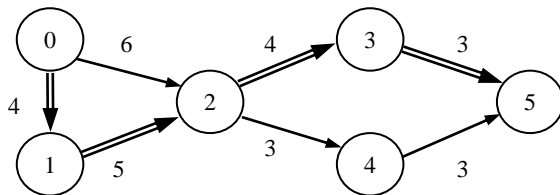


图 G 的关键路径的长度为 16。(1 分)

【评分说明】若考生给出的答案 (1) 或 (2) 不完全正确, 但所画的图是含 6 个顶点、7 条边的有向无环图, 且所求得的关键路径及路径长度与考生所画的图吻合, 则按 (3) 的标准给分。部分正确, 可酌情给分。

42. 【答案要点】

(1) 给出算法的基本设计思想: (5 分)

分别求两个升序序列 A、B 的中位数, 设为 a 和 b。若 a=b, 则 a 或 b 即为所求的中位数; 否则, 舍弃 a、b 中较小者所在序列之较小一半, 同时舍弃较大者所在序列之较大一半, 要求两次舍弃的元素个数相同。在保留的两个升序序列中, 重复上述过程, 直到两个序列中均只含一个元素时为止, 则较小者即为所求的中位数。

(2) 算法实现: (8 分)

```
int M_Search ( int A[], int B[], int n )
{ int start1, end1, mid1, start2, end2, mid2;
  start1 = 0;    end1 = n-1;
  start2 = 0;    end2 = n-1;
```

```

while ( start1 != end1 || start2 != end2 )
{
    mid1 = (start1 + end1) / 2;
    mid2 = (start2 + end2) / 2;
    if ( A[ mid1 ] == B[ mid2 ] )
        return A[ mid1 ];
    if ( A[ mid1 ] < B[ mid2 ] )
    {
        // 分别考虑奇数和偶数，保持两个子数组元素个数相等
        if ( ( start1 + end1 ) % 2 == 0 )
        {
            // 若元素为奇数个
            start1 = mid1;      // 舍弃 A 中间点以前的部分且保留中间点
            end2 = mid2;      // 舍弃 B 中间点以后的部分且保留中间点
        }
        else
        {
            // 若元素为偶数个
            start1 = mid1 + 1; // 舍弃 A 的前半部分
            end2 = mid2;      // 舍弃 B 的后半部分
        }
    }
    else
    {
        if ( ( start1 + end1 ) % 2 == 0 )
        {
            // 若元素为奇数个
            end1 = mid1;      // 舍弃 A 中间点以后的部分且保留中间点
            start2 = mid2;    // 舍弃 B 中间点以前的部分且保留中间点
        }
        else
        {
            // 若元素为偶数个
            end1 = mid1;      // 舍弃 A 的后半部分
            start2 = mid2+1;  // 舍弃 B 的前半部分
        }
    }
}
return A[start1] < B[start2] ? A[start1] : B[start2];
}

```

(3) 上述所给算法的时间、空间复杂度分别是 $O(\log_2 n)$ 和 $O(1)$ 。(2 分)

【评分说明】

①若考生设计的算法满足题目的功能要求且正确，则 (1)、(2) 根据所实现算法的效率给分，细则见下表：

时间	空间	(1)	(2)	说明
----	----	-----	-----	----

复杂度	复杂度	得分	得分	
$O(\log_2 n)$	$O(1)$	5	8	
$O(\log_2 n)$	$O(\log_2 n)$	5	7	如采用递归算法
$O(n)$	$O(1)$	4	7	如采用 2 路归并的思想，程序见表后
$O(n)$	$O(n)$	4	5	如采用 2 路归并的思想，但使用了辅助数组
其他	其他	3	3	如对所有元素进行排序后再查找中位数

// 采用 2 路归并的思想实现

```
int M_Search( int A[], int B[], int n )
```

```
{ int i, j, k;
```

```
  i = j = k = 0;
```

```
  while ( i < n && j < n )
```

```
  {   k++;
```

```
      if ( A[i] < B[j] )
```

```
      {   i++;
```

```
          if ( k == n )
```

```
              return A[i-1];
```

```
      }
```

```
      else
```

```
      {   j++;
```

```
          if ( k == n )
```

```
              return B[j-1];
```

```
      }
```

```
  }
```

```
}
```

②若在算法的基本设计思想描述中因文字表达没有非常清晰反映出算法思路，但在算法实现中能够清晰看出算法思想且正确的，可参照①的标准给分。

③若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。

④若考生所估计的时间复杂度及空间复杂度与考生所实现的算法一致，各给 1 分。

⑤参考答案中只给出了使用 C 语言的版本，使用 C++/JAVA 语言的答案视同使用 C 语言。

2012 年试卷

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。

1. 求整数 n ($n \geq 0$) 阶乘的算法如下，其时间复杂度是

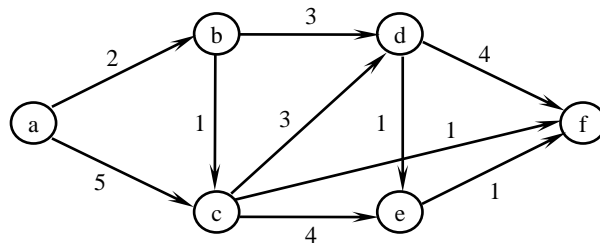
```

int fact( int n )
{
    if ( n <= 1 ) return 1;
    return n * fact( n-1 );
}

```

A. $O(\log_2 n)$ B. $O(n)$ C. $O(n \log_2 n)$ D. $O(n^2)$

2. 已知操作符包括‘+’、‘-’、‘*’、‘/’、‘(’和‘)’. 将中缀表达式 $a+b-a*((c+d)/e-f)+g$ 转换为等价的后缀表达式 $ab+acd+e/f-*g+$ 时, 用栈来存放暂时还不能确定运算次序的操作符. 若栈初始时空, 则转换过程中同时保存在栈中的操作符的最大个数是
A. 5 B. 7 C. 8 D. 11
3. 若一棵二叉树的前序遍历序列为 a, e, b, d, c, 后序遍历序列为 b, c, d, e, a, 则根结点的孩子结点
A. 只有 e B. 有 e、b C. 有 e、c D. 无法确定
4. 若平衡二叉树的高度为 6, 且所有非叶结点的平衡因子均为 1, 则该平衡二叉树的结点总数为
A. 12 B. 20 C. 32 D. 33
5. 对有 n 个顶点、 e 条边且使用邻接表存储的有向图进行广度优先遍历, 其算法时间复杂度是
A. $O(n)$ B. $O(e)$ C. $O(n+e)$ D. $O(n \times e)$
6. 若用邻接矩阵存储有向图, 矩阵中主对角线以下的元素均为零, 则关于该图拓扑序列的结论是
A. 存在, 且唯一 B. 存在, 且不唯一
C. 存在, 可能不唯一 D. 无法确定是否存在
7. 对如下有向带权图, 若采用迪杰斯特拉 (Dijkstra) 算法求从源点 a 到其他各顶点的最短路径, 则得到的第一条最短路径的目标顶点是 b, 第二条最短路径的目标顶点是 c, 后续得到的其余各最短路径的目标顶点依次是



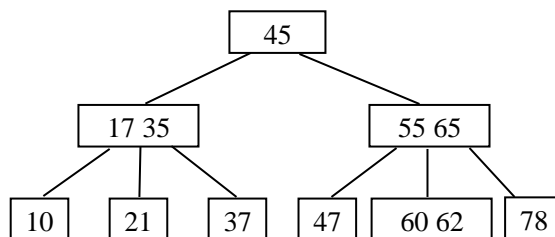
A. d, e, f B. e, d, f C. f, d, e D. f, e, d

8. 下列关于最小生成树的叙述中, 正确的是
I. 最小生成树的代价唯一

- II. 所有权值最小的边一定会出现在所有的最小生成树中
 III. 使用普里姆 (Prim) 算法从不同顶点开始得到的最小生成树一定相同
 IV. 使用普里姆算法和克鲁斯卡尔 (Kruskal) 算法得到的最小生成树总不相同

A. 仅 I B. 仅 II C. 仅 I、III D. 仅 II、IV

9. 设有一棵 3 阶 B 树，如下图所示。删除关键字 78 得到一棵新 B 树，其最右叶结点所含的关键字是



A. 60 B. 60, 62 C. 62, 65 D. 65

10. 排序过程中，对尚未确定最终位置的所有元素进行一遍处理称为一趟排序。下列排序方法中，每一趟排序结束时都至少能够确定一个元素最终位置的方法是

I. 简单选择排序 II. 希尔排序 III. 快速排序
 IV. 堆排序 V. 二路归并排序

A. 仅 I、III、IV B. 仅 I、III、V
 C. 仅 II、III、IV D. 仅 III、IV、V

11. 对同一待排序列分别进行折半插入排序和直接插入排序，两者之间可能的不同之处是

A. 排序的总趟数 B. 元素的移动次数
 C. 使用辅助空间的数量 D. 元素之间的比较次数

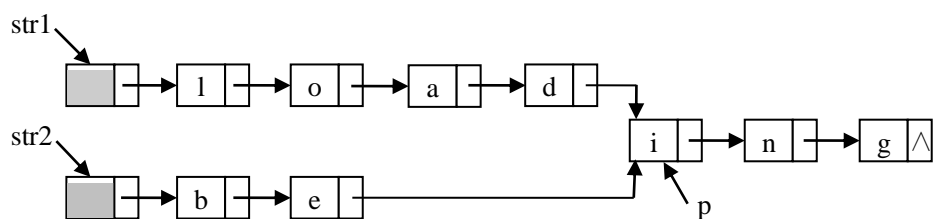
二、综合应用题：41~47 小题，共 70 分。

41. (10 分) 设有 6 个有序表 A、B、C、D、E、F，分别含有 10、35、40、50、60 和 200 个数据元素，各表中元素按升序排列。要求通过 5 次两两合并，将 6 个表最终合并成 1 个升序表，并在最坏情况下比较的总次数达到最小。请回答下列问题。

(1) 给出完整的合并过程，并求出最坏情况下比较的总次数。

(2) 根据你的合并过程，描述 n ($n \geq 2$) 个不等长升序表的合并策略，并说明理由。

42. (13 分) 假定采用带头结点的单链表保存单词，当两个单词有相同的后缀时，则可共享相同的后缀存储空间。例如，“loading”和“being”的存储映像如下图所示。



设 $str1$ 和 $str2$ 分别指向两个单词所在单链表的头结点，链表结点结构为

data	next
------	------

，请设计一个时间上尽可能高效的算法，找出由 $str1$ 和 $str2$ 所指的两个链表共同后缀的起始位置（如图中字符 i 所在结点的位置 p ）。要求：

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想，采用 C 或 C++ 或 JAVA 语言描述算法，关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度。

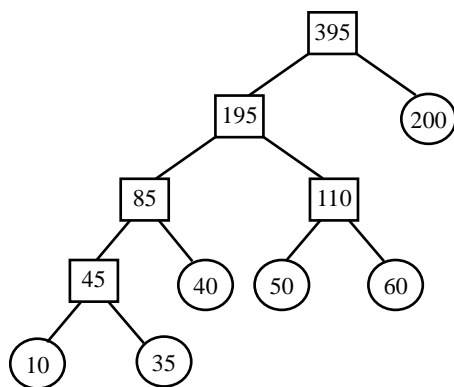
2012 年答案

一、

- | | | | | |
|-------|------|------|------|-------|
| 1. B | 2. A | 3. A | 4. B | 5. C |
| 6. C | 7. C | 8. A | 9. D | 10. A |
| 11. D | | | | |

二、

41. (1) 6 个表的合并顺序如下图所示。



对应于合并过程的哈夫曼树

根据上图中的哈夫曼树，6 个序列的合并过程为：

第 1 次合并：表 A 与表 B 合并，生成含 45 个元素的表 AB；

第 2 次合并：表 AB 与表 C 合并，生成含 85 个元素的表 ABC；

第 3 次合并：表 D 与表 E 合并，生成含 110 个元素的表 DE；

第 4 次合并：表 ABC 与表 DE 合并，生成含 195 个元素的表 ABCDE；

第 5 次合并：表 ABCDE 与表 F 合并，生成含 395 个元素的最终表；

由于合并两个长度分别为 m 和 n 的有序表，最坏情况下需要比较 $m+n-1$ 次，故最坏情况下比较的总次数计算如下：

第 1 次合并：最多比较次数 = $10+35-1 = 44$

第 2 次合并：最多比较次数 = $45+40-1 = 84$

第 3 次合并：最多比较次数 = $50+60-1 = 109$

第 4 次合并：最多比较次数 = $85+110-1 = 194$

第 5 次合并：最多比较次数 = $195+200-1 = 394$

比较的总次数最多为： $44+84+109+194+394 = 825$

(2) 各表的合并策略是：在对多个有序表进行两两合并时，若表长不同，则最坏情况下总的比较次数依赖于表的合并次序。可以借用哈夫曼树的构造思想，依次选择最短的两个表进行合并，可以获得最坏情况下最佳的合并效率。

42.

(1)

①分别求出 $str1$ 和 $str2$ 所指的链表的长度 m 和 n ；

②将两个链表以表尾对齐：令指针 p 、 q 分别指向 $str1$ 和 $str2$ 的头结点，若 $m \geq n$ ，则使 p 指向链表中的第 $m - n + 1$ 个结点；若 $m < n$ ，则使 q 指向链表中的第 $n - m + 1$ 个结点，即使指针 p 和 q 所指的结点到表尾的长度相等。

③反复将指针 p 和 q 同步向后移动，并判断它们是否指向同一结点。若 p 和 q 指向同一结点，则该点即为所求的共同后缀的起始位置。

(2) 算法实现

```
typedef struct Node
```

```
{
```

```
    char data;
```

```
    struct Node * next;
```

```
} SNode;
```

```
SNode * findlist( SNode * str1, SNode * str2 )
```

```
{
```

```
    int m, n;
```

```
    SNode * p, * q;
```

```
    m = listlen( str1 );           /* 求 str1 的长度。O(m) */
```

```
    n = listlen( str2 );           /* 求 str2 的长度。O(n) */
```

(3) 参考答案的时间复杂度为： $O(m+n)$ 或 $O(\max(m,n))$ 。其中 m 、 n 分别为两个链表的长度。

子为 0 的分支结点的个数是

- A. 0 B. 1 C. 2 D. 3

4. 已知三叉树 T 中 6 个叶结点的权分别是 2, 3, 4, 5, 6, 7, T 的带权 (外部) 路径长度最小是

- A. 27 B. 46 C. 54 D. 56

5. 若 X 是后序线索二叉树中的叶结点, 且 X 存在左兄弟结点 Y , 则 X 的右线索指向的是

- A. X 的父结点 B. 以 Y 为根的子树的最左下结点
C. X 的左兄弟结点 Y D. 以 Y 为根的子树的最右下结点

6. 在任意一棵非空二叉排序树 T_1 中, 删除某结点 v 之后形成二叉排序树 T_2 , 再将 v 插入 T_2 形成二叉排序树 T_3 。下列关于 T_1 与 T_3 的叙述中, 正确的是

- I. 若 v 是 T_1 的叶结点, 则 T_1 与 T_3 不同
II. 若 v 是 T_1 的叶结点, 则 T_1 与 T_3 相同
III. 若 v 不是 T_1 的叶结点, 则 T_1 与 T_3 不同
IV. 若 v 不是 T_1 的叶结点, 则 T_1 与 T_3 相同

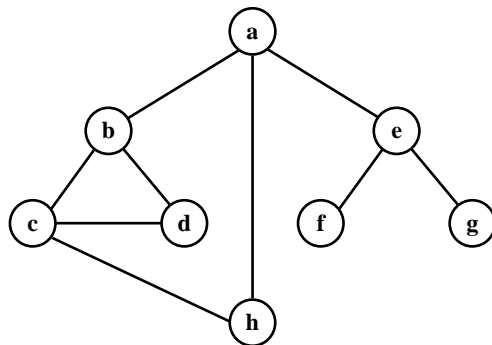
- A. 仅 I、III B. 仅 I、IV
C. 仅 II、III D. 仅 II、IV

7. 设图的邻接矩阵 A 如下所示。各顶点的度依次是

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

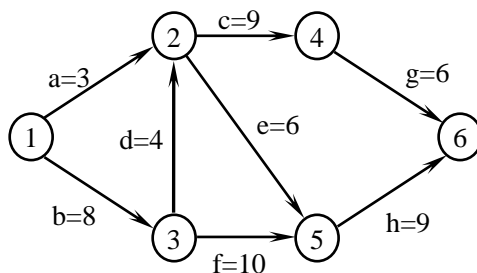
- A. 1, 2, 1, 2 B. 2, 2, 1, 1
C. 3, 4, 2, 3 D. 4, 4, 2, 2

8. 若对如下无向图进行遍历, 则下列选项中, 不是广度优先遍历序列的是



- A. h, c, a, b, d, e, g, f B. e, a, f, g, b, h, c, d
C. d, b, c, a, h, e, f, g D. a, b, c, d, h, e, f, g

9. 下列 AOE 网表示一项包含 8 个活动的工程。通过同时加快若干活动的进度，可以缩短整个工程的工期。下列选项中，加快其进度就可以缩短工程工期的是



- A. c 和 e
B. d 和 c
C. f 和 d
D. f 和 h
10. 在一棵高度为 2 的 5 阶 B 树中，所含关键字的个数最少是
A. 5
B. 7
C. 8
D. 14
11. 对给定的关键字序列 110, 119, 007, 911, 114, 120, 122 进行基数排序，则第 2 趟分配收集后得到的关键字序列是
A. 007, 110, 119, 114, 911, 120, 122
B. 007, 110, 119, 114, 911, 122, 120
C. 007, 110, 911, 114, 119, 120, 122
D. 110, 120, 911, 122, 114, 007, 119

2013 年答案

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项符合试题要求。请在答题卡上将所选项的字母涂黑。

1. D 2. C 3. D 4. B 5. A
6. C 7. C 8. D 9. C 10. A
11. C

二、综合应用题：41~47 小题，共 70 分。请将答案写在答题卡指定位置的边框区域内。

41. 【答案要点】

(1) 给出算法的基本设计思想：(4 分)

算法的策略是从前向后扫描数组元素，标记出一个可能成为主元素的元素 Num。然后重新计数，确认 Num 是否是主元素。

算法可分为以下两步：

- ① 选取候选的主元素：依次扫描所给数组中的每个整数，将第一个遇到的整数 Num 保存到 c 中，记录 Num 的出现次数为 1；若遇到的下一个整数仍等于 Num，则计数加 1，否则计数减 1；当计数减到 0 时，将遇到的下一个整数保存到 c 中，计数重新记为 1，开始新一轮计数，即从当前位置开始重复上述过程，直到扫描完全部数组元素。
- ② 判断 c 中元素是否是真正的主元素：再次扫描该数组，统计 c 中元素出现的次

数，若大于 $n/2$ ，则为主元素，否则，序列中不存在主元素。

(2) 算法实现：(7 分)

```
int Majority( int A[], int n )
{
    int i, c, count=1;           // c 用来保存候选主元素，count 用来计数
    c = A[0];                    // 设置 A[0]为候选主元素
    for ( i=1; i<n; i++ )        // 查找候选主元素
        if ( A[i] == c )
            count++;              // 对 A 中的候选主元素计数
        else
            if (count > 0)        // 处理不是候选主元素的情况
                count--;
            else                  // 更换候选主元素，重新计数
                {
                    c = A[i];
                    count = 1;
                }
    if ( count>0 )
        for ( i=count=0; i<n; i++ )    // 统计候选主元素的实际出现次数
            if ( A[i] == c )
                count++;
    if ( count > n/2 )            // 确认候选主元素
        return c;
    else
        return -1;              // 不存在主元素
}
```

【(1) (2) 的评分说明】

① 若考生设计的算法满足题目的功能要求且正确，则 (1)、(2) 根据所实现算法的效率给分，细则见下表：

时间 复杂度	空间 复杂度	(1) 得分	(2) 得分	说明
$O(n)$	$O(1)$	4	7	
$O(n)$	$O(n)$	4	6	如采用计数排序思想，见表后 Majority1 程序
$O(n\log_2 n)$	其他	3	6	如采用其他排序的思想
$\geq O(n^2)$	其他	3	5	其他方法

int Majority1(int A[], int n) // 采用计数排序思想，时间： $O(n)$ ，空间： $O(n)$

```
{
    int k, *p, max;
    p = (int *) malloc( sizeof(int)*n); // 申请辅助计数数组
    for ( k=0; k < n; k++ )            // 计数数组清 0
        p[k] = 0;
    max = 0;
```

```

for ( k=0; k<n; k++ )
{
    p[ A[k] ] ++;                // 计数器+1
    if ( p[ A[k] ] > p[ max ] )
        max = A[k];              // 记录出现次数最多的元素
}
if ( p[max] > n/2 )
    return max;
else
    return -1;
}

```

- ② 若在算法的基本设计思想描述中因文字表达没有非常清晰反映出算法思路，但在算法实现中能够清晰看出算法思想且正确的，可参照①的标准给分。
- ③ 若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。
- ④ 参考答案中只给出了使用 C 语言的版本，使用 C++ 或 Java 语言的答案视同使用 C 语言。

(3) 说明算法复杂性：(2 分)

参考答案中实现的程序的时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。

【评分说明】若考生所估计的时间复杂度与空间复杂度与考生所实现的算法一致，可各给 1 分。

42. 【答案要点】

(1) 采用顺序存储结构，数据元素按其查找概率降序排列。(2 分)

采用顺序查找方法。(1 分)

查找成功时的平均查找长度 = $0.35 \times 1 + 0.35 \times 2 + 0.15 \times 3 + 0.15 \times 4 = 2.1$ 。(2 分)

(2)

【答案一】

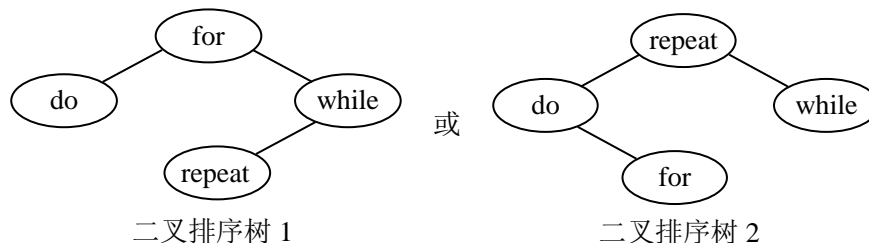
采用链式存储结构，数据元素按其查找概率降序排列，构成单链表。(2 分)

采用顺序查找方法。(1 分)

查找成功时的平均查找长度 = $0.35 \times 1 + 0.35 \times 2 + 0.15 \times 3 + 0.15 \times 4 = 2.1$ 。(2 分)

【答案二】

采用二叉链表存储结构，构造二叉排序树，元素存储方式见下图。(2 分)



采用二叉排序树的查找方法。(1 分)

查找成功时的平均查找长度= $0.15 \times 1 + 0.35 \times 2 + 0.35 \times 2 + 0.15 \times 3 = 2.0$ 。(2 分)

【(1) (2) 的评分说明】

- ① 若考生以实际元素表示“降序排列”，同样给分。
- ② 若考生正确求出与其查找方法对应的查找成功时的平均查找长度，给 2 分；若计算过程正确，但结果错误，给 1 分。
- ③ 若考生给出其他更高效的查找方法且正确，可参照评分标准给分。

2014 年试卷

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项符合试题要求。请在答题卡上将所选项的字母涂黑。

1. 下列程序段的时间复杂度是

```
count = 0;
for ( k=1; k<=n; k*=2 )
    for ( j=1; j<=n; j++ )
        count++;
```

- A. $O(\log_2 n)$ B. $O(n)$ C. $O(n \log_2 n)$ D. $O(n^2)$

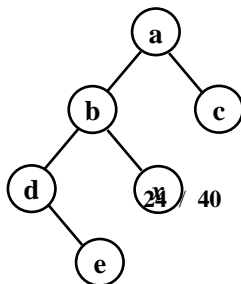
2. 假设栈初始为空，将中缀表达式 $a/b + (c*d - e*f) / g$ 转换为等价的后缀表达式的过程中，当扫描到 f 时，栈中的元素依次是

- A. $+ (* -$ B. $+ (- *$ C. $/ + (* - *$ D. $/ + - *$

3. 循环队列存放在一维数组 $A[0..M-1]$ 中， $end1$ 指向队头元素， $end2$ 指向队尾元素的后一个位置。假设队列两端均可进行入队和出队操作，队列中最多能容纳 $M-1$ 个元素，初始时为空。下列判断队空和队满的条件中，正确的是

- A. 队空： $end1 == end2$ ；队满： $end1 == (end2 + 1) \bmod M$
B. 队空： $end1 == end2$ ；队满： $end2 == (end1 + 1) \bmod (M-1)$
C. 队空： $end2 == (end1 + 1) \bmod M$ ；队满： $end1 == (end2 + 1) \bmod M$
D. 队空： $end1 == (end2 + 1) \bmod M$ ；队满： $end2 == (end1 + 1) \bmod (M-1)$

4. 若对如下的二叉树进行中序线索化，则结点 x 的左、右线索指向的结点分别是



- A. e、c B. e、a C. d、c D. b、a

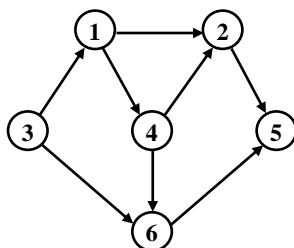
5. 将森林 F 转换为对应的二叉树 T , F 中叶结点的个数等于

- A. T 中叶结点的个数 B. T 中度为 1 的结点个数
C. T 中左孩子指针为空的结点个数 D. T 中右孩子指针为空的结点个数

6. 5 个字符有如下 4 种编码方案, 不是前缀编码的是

- A. 01, 0000, 0001, 001, 1 B. 011, 000, 001, 010, 1
C. 000, 001, 010, 011, 100 D. 0, 100, 110, 1110, 1100

7. 对如下所示的有向图进行拓扑排序, 得到的拓扑序列可能是



- A. 3, 1, 2, 4, 5, 6 B. 3, 1, 2, 4, 6, 5
C. 3, 1, 4, 2, 5, 6 D. 3, 1, 4, 2, 6, 5

8. 用哈希 (散列) 方法处理冲突 (碰撞) 时可能出现堆积 (聚集) 现象。下列选项中, 会受堆积现象直接影响的是

- A. 存储效率 B. 散列函数
C. 装填 (装载) 因子 D. 平均查找长度

9. 在一棵具有 15 个关键字的 4 阶 B 树中, 含关键字的结点个数最多是

- A. 5 B. 6 C. 10 D. 15

10. 用希尔排序方法对一个数据序列进行排序时, 若第 1 趟排序结果为 9, 1, 4, 13, 7, 8, 20, 23, 15, 则该趟排序采用的增量 (间隔) 可能是

- A. 2 B. 3 C. 4 D. 5

11. 下列选项中, 不可能是快速排序第 2 趟排序结果的是

- A. 2, 3, 5, 4, 6, 7, 9 B. 2, 7, 5, 6, 4, 3, 9
C. 3, 2, 5, 4, 7, 6, 9 D. 4, 2, 3, 5, 7, 6, 9

二、综合应用题: 41~47 小题, 共 70 分。请将答案写在答题卡指定位置的边框区域内。

41. (13 分) 二叉树的带权路径长度 (WPL) 是二叉树中所有叶结点的带权路径长度之和。给定一棵二叉树 T , 采用二叉链表存储, 结点结构为:

left	weight	right
------	--------	-------

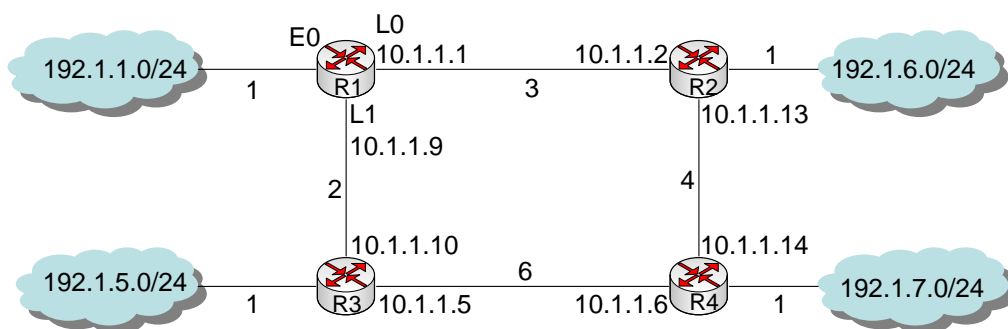
, 其中叶结点的 weight 域保存该结点的非负权值。设 root 为指向 T 的根结点的指针, 请设计求 T 的 WPL 的算法。要求:

- (1) 给出算法的基本设计思想；
- (2) 使用 C 或 C++ 语言，给出二叉树结点的数据类型定义；
- (3) 根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。

42. (10 分) 某网络中的路由器运行 OSPF 路由协议，题 42 表是路由器 R1 维护的主要链路状态信息 (LSI)，题 42 图是根据题 42 表及 R1 的接口名构造出来的网络拓扑。

题 42 表 R1 所维护的 LSI

		R1 的 LSI	R2 的 LSI	R3 的 LSI	R4 的 LSI	备注
Router ID		10.1.1.1	10.1.1.2	10.1.1.5	10.1.1.6	标识路由器的 IP 地址
Link1	ID	10.1.1.2	10.1.1.1	10.1.1.6	10.1.1.5	所连路由器的 Router ID
	IP	10.1.1.1	10.1.1.2	10.1.1.5	10.1.1.6	Link1 的本地 IP 地址
	Metric	3	3	6	6	Link1 的费用
Link2	ID	10.1.1.5	10.1.1.6	10.1.1.1	10.1.1.2	所连路由器的 Router ID
	IP	10.1.1.9	10.1.1.13	10.1.1.10	10.1.1.14	Link2 的本地 IP 地址
	Metric	2	4	2	4	Link2 的费用
Net1	Prefix	192.1.1.0/24	192.1.6.0/24	192.1.5.0/24	192.1.7.0/24	直连网络 Net1 的网络前缀
	Metric	1	1	1	1	到达直连网络 Net1 的费用



题 42 图 R1 构造的网络拓扑

请回答下列问题。

- (1) 本题中的网络可抽象为数据结构中的哪种逻辑结构？
- (2) 针对题 42 表中的内容，设计合理的链式存储结构，以保存题 42 表中的链路状态信息 (LSI)。要求给出链式存储结构的数据类型定义，并画出对应题 42 表的链式存储结构示意图 (示意图中可仅以 ID 标识结点)。
- (3) 按照迪杰斯特拉 (Dijkstra) 算法的策略，依次给出 R1 到达题 42 图中子网 192.1.x.x 的最短路径及费用。

2014 年答案

一、单项选择题：每小题 2 分，共 80 分。

1. C 2. B 3. A 4. D 5. C

(3) 算法实现 (7 分)

```
int WPL( BTree * root )           // 基于递归的后序遍历算法实现
{
    int w_l, w_r;
    if ( root->left == NULL && root->right == NULL )
        return 0;
    else
    {
        w_l = WPL( root->left );      // 计算左子树的 WPL
        w_r = WPL( root->right );     // 计算右子树的 WPL
        root->weight = root->left->weight + root->right->weight;
                                     // 填写非叶结点的 weight 域
        return ( w_l + w_r + root->weight ); // 返回 WPL 值
    }
}
```

【评分说明】

- ① 若考生给出能够满足题目要求的其他算法（包括用非递归的遍历方式实现的算法），且正确，可同样给分。
- ② 参考答案中只给出了使用 C 语言的版本，使用 C++ 语言正确实现的算法同样给分。
- ③ 若对算法的基本设计思想和主要数据结构的描述不十分清晰，但在算法实现中能够清晰反映出算法思想且正确，参照①的标准给分。
- ④ 若考生给出的二叉树结点的数据类型定义及算法实现中，使用的是除整型之外的其他数值类型，可视同使用整型类型。
- ⑤ 若考生给出的答案中算法主要设计思想或算法实现中部分正确，可酌情给分。

42. 【答案要点】

(1) 本题中的网络可抽象为图结构。(1 分)

【评分说明】

只要考生的答案中给出与“图”的含义相似的描述，例如“网状结构”、“非线性结构”等，同样给分。

(2) 链式存储结构的数据类型定义如下：(3 分)

```
typedef struct
{
    unsigned int  ID;
    unsigned int  IP;
} LinkNode;      // Link 的结构

typedef struct
{
    unsigned int  Prefix;
```

```

    unsigned int  Mask;
} NetNode;          // Net 的结构

```

```

typedef struct Node

```

```

{   int  Flag;      // Flag=1: Link; Flag=2: Net

```

```

    union

```

```

    {   LinkNode  Lnode;

```

```

        NetNode  Nnode;

```

```

    } LinkORNet;

```

```

    unsigned int  Metric;

```

```

    struct Node  *Next;

```

```

} ArcNode;          // 弧结点

```

弧结点的两种基本形态

Flag=1	Next	Flag=2	Next
ID		Prefix	
IP		Mask	
Metric		Metric	

```

typedef struct HNode

```

```

{   unsigned int  RouterID;

```

```

    ArcNode      *LN_link;

```

```

    struct HNode *Next;

```

```

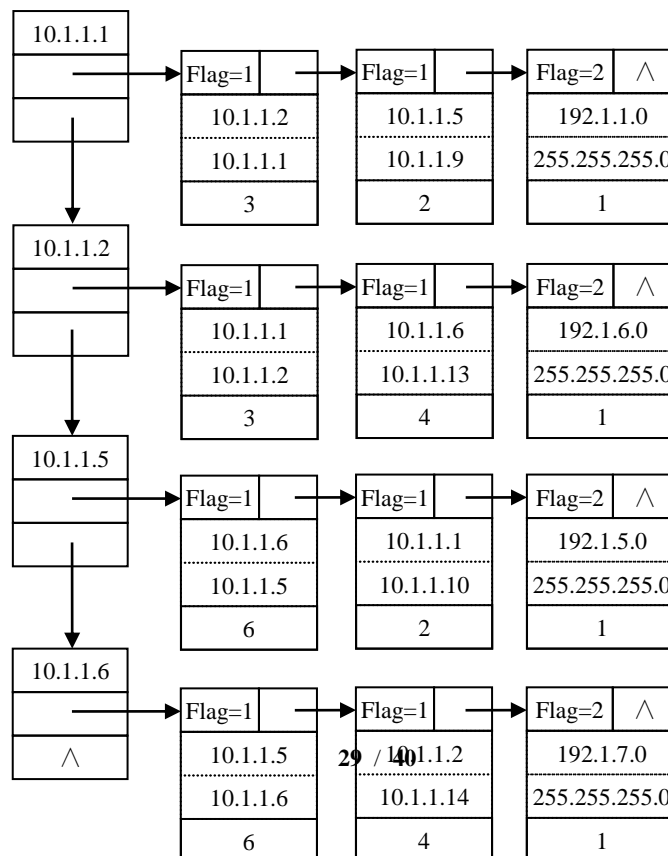
} HNODE;          // 表头结点

```

表头结点
结构示意图

RouterID
LN_link
Next

对应题 42 表的链式存储结构示意图如下。(2 分)



【评分说明】

- ① 若考生给出的答案是将链表中的表头结点保存在一个一维数组中(即采用邻接表形式), 同样给分。
- ② 若考生给出的答案中, 弧结点定义没有使用 **union**, 而是采用两种不同的结构分别表示 **Link** 和 **Net**, 同时在表头结点中定义了两个指针, 分别指向由这两种类型的结点构成的两个链表, 同样给分。
- ③ 考生所给答案的弧结点中, 可以在单独定义的域中保存各直连网络 IP 地址的前缀长度, 也可以与网络地址保存在同一个域中。
- ④ 数据类型定义中, 只要采用了可行的链式存储结构, 并保存了题目中所给的 LSI 信息, 例如将网络抽象为一类结点, 写出含 8 个表头结点的链式存储结构, 均可参照①~③的标准给分。
- ⑤ 考生给出的答案中, 图示部分应与其数据类型定义部分一致, 图示只要能够体现链式存储结构及题 42 图中的网络连接关系(可以不给出结点内的细节信息), 即可给分。
- ⑥ 若解答不完全正确, 酌情给分。
- ⑦ 参考答案中只给出了使用 C 语言描述的数据类型定义, 若考生使用 C++ 语言解答可参照①~⑥的标准给分。

(3) 计算结果如下表所示。(4 分)

	目的网络	最短路径	代价(费用)
步骤 1	192.1.1.0/24	直接到达	1
步骤 2	192.1.5.0/24	R1→R3→192.1.5.0/24	3
步骤 3	192.1.6.0/24	R1→R2→192.1.6.0/24	4
步骤 4	192.1.7.0/24	R1→R2→R4→192.1.7.0/24	8

【评分说明】

- ① 若考生给出的各条最短路径的结果部分正确, 可酌情给分。
- ② 若考生给出的从 R1 到达子网 192.1.x.x 的最短路径及代价正确, 但不完全符合代价不减的次序, 可酌情给分。

2015 年试卷

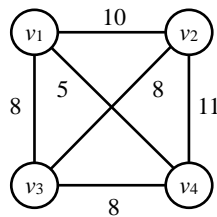
一、单项选择题: 1~40 小题, 每小题 2 分, 共 80 分。下列每题给出的四个选项中, 只有一个选项符合题目要求。请在答题卡上将所选项的字母涂黑。

1. 已知程序如下:

```
int S( int n )
{   return ( n<=0 ) ? 0 : S(n-1) + n;   }
void main( )
{   cout << S(1); }
```

程序运行时使用栈来保存调用过程的信息, 自栈底到栈顶保存的信息依次对应的是

- A. $\text{main}() \rightarrow S(1) \rightarrow S(0)$ B. $S(0) \rightarrow S(1) \rightarrow \text{main}()$
 C. $\text{main}() \rightarrow S(0) \rightarrow S(1)$ D. $S(1) \rightarrow S(0) \rightarrow \text{main}()$
2. 先序序列为 a, b, c, d 的不同二叉树的个数是
 A. 13 B. 14 C. 15 D. 16
3. 下列选项给出的是从根分别到达两个叶结点路径上的权值序列，能属于同一棵哈夫曼树的是
 A. 24, 10, 5 和 24, 10, 7 B. 24, 10, 5 和 24, 12, 7
 C. 24, 10, 10 和 24, 14, 11 D. 24, 10, 5 和 24, 14, 6
4. 现有一棵无重复关键字的平衡二叉树 (AVL 树)，对其进行中序遍历可得到一个降序序列。下列关于该平衡二叉树的叙述中，正确的是
 A. 根结点的度一定为 2 B. 树中最小元素一定是叶结点
 C. 最后插入的元素一定是叶结点 D. 树中最大元素一定无左子树
5. 设有向图 $G = (V, E)$ ，顶点集 $V = \{v_0, v_1, v_2, v_3\}$ ，边集 $E = \{ \langle v_0, v_1 \rangle, \langle v_0, v_2 \rangle, \langle v_0, v_3 \rangle, \langle v_1, v_3 \rangle \}$ 。若从顶点 v_0 开始对图进行深度优先遍历，则可能得到的不同遍历序列个数是
 A. 2 B. 3 C. 4 D. 5
6. 求下面带权图的最小 (代价) 生成树时，可能是克鲁斯卡尔 (Kruskal) 算法第 2 次选中但不是普里姆 (Prim) 算法 (从 v_4 开始) 第 2 次选中的边是



- A. (v_1, v_3) B. (v_1, v_4) C. (v_2, v_3) D. (v_3, v_4)
7. 下列选项中，不能构成折半查找中关键字比较序列的是
 A. 500, 200, 450, 180 B. 500, 450, 200, 180
 C. 180, 500, 200, 450 D. 180, 200, 500, 450
8. 已知字符串 s 为 “abaabaabacacaabaabcc”，模式串 t 为 “abaabc”。采用 KMP 算法进行匹配，第一次出现 “失配” ($s[i] \neq t[j]$) 时， $i = j = 5$ ，则下次开始匹配时， i 和 j 的值分别是
 A. $i = 1, j = 0$ B. $i = 5, j = 0$ C. $i = 5, j = 2$ D. $i = 6, j = 2$
9. 下列排序算法中，元素的移动次数与关键字的初始排列次序无关的是
 A. 直接插入排序 B. 起泡排序 C. 基数排序 D. 快速排序
10. 已知小根堆为 8, 15, 10, 21, 34, 16, 12，删除关键字 8 之后需重建堆，在此过程中，

关键字之间的比较次数是

- A. 1 B. 2 C. 3 D. 4

11. 希尔排序的组内排序采用的是

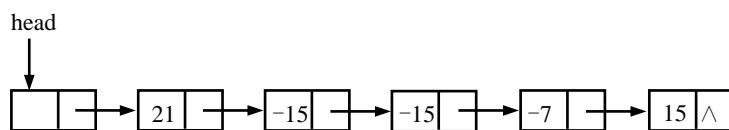
- A. 直接插入排序 B. 折半插入排序 C. 快速排序 D. 归并排序

二、综合应用题：41~47 小题，共 70 分。请将答案写在答题卡指定位置的边框区域内。

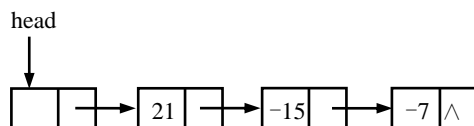
41. (15 分) 用单链表保存 m 个整数，结点的结构为：

data	link
------	------

，且 $|\text{data}| \leq n$ (n 为正整数)。现要求设计一个时间复杂度尽可能高效的算法，对于链表中 data 的绝对值相等的结点，仅保留第一次出现的结点而删除其余绝对值相等的结点。例如，若给定的单链表 head 如下：



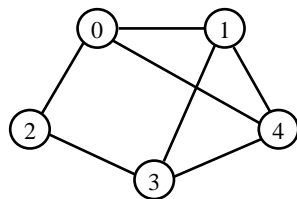
则删除结点后的 head 为：



要求：

- (1) 给出算法的基本设计思想。
- (2) 使用 C 或 C++ 语言，给出单链表结点的数据类型定义。
- (3) 根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。
- (4) 说明你所设计算法的时间复杂度和空间复杂度。

42. (8 分) 已知含有 5 个顶点的图 G 如下图所示。



请回答下列问题。

- (1) 写出图 G 的邻接矩阵 A (行、列下标均从 0 开始)。
- (2) 求 A^2 ，矩阵 A^2 中位于 0 行 3 列元素值的含义是什么？
- (3) 若已知具有 n ($n \geq 2$) 个顶点的图的邻接矩阵为 B ，则 B^m ($2 \leq m \leq n$) 中非零元素的含义是什么？

2015 年答案

一、单项选择题：每小题 2 分，共 80 分。

- | | | | | |
|-------|------|------|------|-------|
| 1. A | 2. B | 3. D | 4. D | 5. D |
| 6. C | 7. A | 8. C | 9. C | 10. C |
| 11. A | | | | |

二、综合应用题：41~47 小题，共 70 分。

41. 【答案要点】

(1) 算法的基本设计思想 (4 分)

算法的核心思想是用空间换时间。使用辅助数组记录链表中已出现的数值，从而只需对链表进行一趟扫描。

因为 $|data| \leq n$ ，故辅助数组 q 的大小为 $n+1$ ，各元素的初值均为 0。依次扫描链表中的各结点，同时检查 $q[|data|]$ 的值，如果为 0，则保留该结点，并令 $q[|data|]=1$ ；否则，将该结点从链表中删除。

(2) 使用 C 语言描述的单链表结点的数据类型定义 (2 分)

```
typedef struct node {
    int      data;
    struct node *link;
} NODE;
typedef NODE *PNODE;
```

(3) 算法实现 (7 分)

```
void func( PNODE h, int n )
{
    PNODE p=h, r;
    int *q, m;

    q = (int *) malloc( sizeof(int)*(n+1));    // 申请 n+1 个位置的辅助空间
    for ( int i=0; i<n+1; i++ )                // 数组元素初值置 0
        *(q+i) = 0;

    while ( p->link != NULL )
    {
        m = p->link->data>0 ? p->link->data : -p->link->data;
        if ( *(q+m) == 0 )                    // 判断该结点的 data 是否已出现过
        {
            *(q+m) = 1;                        // 首次出现
            p = p->link;                        // 保留
        }
        else                                  // 重复出现
        {
            r = p->link;                        // 删除
            p = r;
        }
    }
}
```

```

        p->link = r->link;
        free( r );
    }
}
free( q );
}

```

【(1)、(2)、(3) 的评分说明】

- ① 若考生设计的算法满足题目的功能要求且正确，则 (1)、(3) 根据所实现算法的时间复杂度给分，细则见下表：

时间复杂度	分数	说明
$O(m)$	11	对链表进行一趟扫描，且采用时间复杂度为 $O(1)$ 的方法判断 $ data $ 是否是首次出现。
$O(m)$	8	对链表进行一趟扫描，且采用时间复杂度 $>O(1)$ 的方法判断 $ data $ 是否是首次出现。
$>O(m)$	5	对链表进行多趟扫描，例如采用顺序查找算法，对每个结点在当前链表中进行查找，删除重复出现的结点。

- ② 若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路，但在算法实现中能够表达出算法思想且正确的，可参照①的标准给分。
- ③ 若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。
- ④ 若考生给出的单链表结点的数据类型定义与题目中所给的结点形式不完全相同，酌情给分。
- ⑤ 参考答案中只给出了使用 C 语言的版本，使用 C++ 语言的答案视同使用 C 语言。
- (4) 参考答案所给算法的时间复杂度为 $O(m)$ ，(1 分) 空间复杂度为 $O(n)$ 。(1 分)

【评分说明】若考生所估计的时间复杂度和空间复杂度与考生实现的算法一致，可各给 1 分。

42. 【答案要点】

- (1) 图 G 的邻接矩阵 A 如下：(2 分)

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- (2) A^2 如下：(2 分)

$$A^2 = \begin{bmatrix} 3 & 1 & 0 & 3 & 1 \\ 1 & 3 & 2 & 1 & 2 \\ 0 & 2 & 2 & 0 & 2 \\ 3 & 1 & 0 & 3 & 1 \\ 1 & 2 & 2 & 1 & 3 \end{bmatrix}$$

- 0 行 3 列的元素值 3 表示从顶点 0 到顶点 3 之间长度为 2 的路径共有 3 条。(2 分)
- (3) B^m ($2 \leq m \leq n$) 中位于 i 行 j 列 ($0 \leq i, j \leq n-1$) 的非零元素的含义是：图中从顶点 i 到顶点 j 长度为 m 的路径条数。(2 分)

【评分说明】

- ① 若考生给出的邻接矩阵 A 中，结点与行、列的对应次序与本参考答案不完全一致，只要正确，同样给分。问题 (2) 中，考生所给的答案中顶点编号要与其所给的邻接矩阵相对应。
- ② 若考生给出的矩阵 A 及 A^2 部分正确，酌情给分。
- ③ 若考生分别说明矩阵 B^2 、 B^3 、 \dots 、 B^n 中非零元素的含义，同样给分。
- ④ 若考生给出的 B^m ($2 \leq m \leq n$) 中非零元素的含义部分正确，酌情给分。

2016 年试卷

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项符合试题要求。请在答题卡上将所选项的字母涂黑。

1. 已知表头元素为 c 的单链表在内存中的存储状态如下表所示。

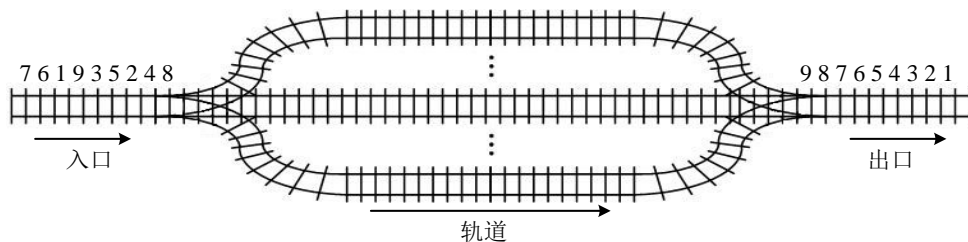
地址	元素	链接地址
1000H	a	1010H
1004H	b	100CH
1008H	c	1000H
100CH	d	NULL
1010H	e	1004H
1014H		

现将 f 存放于 1014H 处并插入到单链表中，若 f 在逻辑上位于 a 和 e 之间，则 a, e, f 的“链接地址”依次是

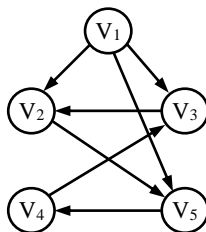
- A. 1010H, 1014H, 1004H B. 1010H, 1004H, 1014H
- C. 1014H, 1010H, 1004H D. 1014H, 1004H, 1010H
2. 已知一个带有表头结点的双向循环链表 L ，结点结构为

prev	data	next
------	------	------

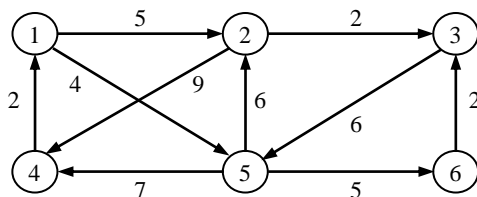
，其中， $prev$ 和 $next$ 分别是指向其直接前驱和直接后继结点的指针。现要删除指针 p 所指的结点，正确的语句序列是
- A. $p \rightarrow next \rightarrow prev = p \rightarrow prev$; $p \rightarrow prev \rightarrow next = p \rightarrow prev$; $free(p)$;
- B. $p \rightarrow next \rightarrow prev = p \rightarrow next$; $p \rightarrow prev \rightarrow next = p \rightarrow next$; $free(p)$;
- C. $p \rightarrow next \rightarrow prev = p \rightarrow next$; $p \rightarrow prev \rightarrow next = p \rightarrow prev$; $free(p)$;
- D. $p \rightarrow next \rightarrow prev = p \rightarrow prev$; $p \rightarrow prev \rightarrow next = p \rightarrow next$; $free(p)$;
3. 设有如下图所示的火车车轨，入口到出口之间有 n 条轨道，列车的行进方向均为从左至右，列车可驶入任意一条轨道。现有编号为 1~9 的 9 列列车，驶入的次序依次是 8, 4, 2, 5, 3, 9, 1, 6, 7。若期望驶出的次序依次为 1 至 9，则 n 至少是



- A. 2 B. 3 C. 4 D. 5
4. 有一个 100 阶的三对角矩阵 M , 其元素 $m_{i,j}$ ($1 \leq i \leq 100, 1 \leq j \leq 100$) 按行优先次序压缩存入下标从 0 开始的一维数组 N 中。元素 $m_{30,30}$ 在 N 中的下标是
- A. 86 B. 87 C. 88 D. 89
5. 若森林 F 有 15 条边、25 个结点, 则 F 包含树的个数是
- A. 8 B. 9 C. 10 D. 11
6. 下列选项中, 不是下图深度优先搜索序列的是



- A. V_1, V_5, V_4, V_3, V_2 B. V_1, V_3, V_2, V_5, V_4
- C. V_1, V_2, V_5, V_4, V_3 D. V_1, V_2, V_3, V_4, V_5
7. 若将 n 个顶点 e 条弧的有向图采用邻接表存储, 则拓扑排序算法的时间复杂度是
- A. $O(n)$ B. $O(n+e)$ C. $O(n^2)$ D. $O(n \times e)$
8. 使用迪杰斯特拉 (Dijkstra) 算法求下图中从顶点 1 到其他各顶点的最短路径, 依次得到的各最短路径的目标顶点是



- A. 5, 2, 3, 4, 6 B. 5, 2, 3, 6, 4 C. 5, 2, 4, 3, 6 D. 5, 2, 6, 3, 4
9. 在有 n ($n > 1000$) 个元素的升序数组 A 中查找关键字 x 。查找算法的伪代码如下所示。
- ```

k = 0;
while (k < n 且 A[k] < x) k = k+3;
if (k < n 且 A[k] == x) 查找成功;
else if (k-1 < n 且 A[k-1] == x) 查找成功;

```

else if (k-2 < n 且 A[k-2] == x) 查找成功;  
else 查找失败;

本算法与折半查找算法相比,有可能具有更少比较次数的情形是

- A. 当  $x$  不在数组中  
B. 当  $x$  接近数组开头处  
C. 当  $x$  接近数组结尾处  
D. 当  $x$  位于数组中间位置

10. B+树不同于 B 树的特点之一是

- A. 能支持顺序查找  
B. 结点中含有关键字  
C. 根结点至少有两个分支  
D. 所有叶结点都在同一层上

11. 对 10 TB 的数据文件进行排序,应使用的方法是

- A. 希尔排序  
B. 堆排序  
C. 快速排序  
D. 归并排序

二、综合应用题: 41~47 小题,共 70 分。请将答案写在答题卡指定位置的边框区域内。

42. (8 分) 如果一棵非空  $k(k \geq 2)$  叉树  $T$  中每个非叶结点都有  $k$  个孩子,则称  $T$  为正则  $k$  叉树。请回答下列问题并给出推导过程。

- (1) 若  $T$  有  $m$  个非叶结点,则  $T$  中的叶结点有多少个?  
(2) 若  $T$  的高度为  $h$  (单结点的树  $h=1$ ),则  $T$  的结点数最多为多少个?最少为多少个?

43. (15 分) 已知由  $n(n \geq 2)$  个正整数构成的集合  $A = \{a_k \mid 0 \leq k < n\}$ ,将其划分为两个不相交的子集  $A_1$  和  $A_2$ ,元素个数分别是  $n_1$  和  $n_2$ ,  $A_1$  和  $A_2$  中元素之和分别为  $S_1$  和  $S_2$ 。设计一个尽可能高效的划分算法,满足  $|n_1 - n_2|$  最小且  $|S_1 - S_2|$  最大。要求:

- (1) 给出算法的基本设计思想。  
(2) 根据设计思想,采用 C 或 C++ 语言描述算法,关键之处给出注释。  
(3) 说明你所设计算法的平均时间复杂度和空间复杂度。

## 2016 年答案

一、单项选择题: 每小题 2 分,共 80 分。

1. D                  2. D                  3. C                  4. B                  5. C  
6. D                  7. B                  8. B                  9. B                  10. A  
11. D

二、综合应用题: 41~47 小题,共 70 分。

42. 【答案要点】

- (1) 根据定义,正则  $k$  叉树中仅含有两类结点: 叶结点 (个数记为  $n_0$ ) 和度为  $k$  的分支结点 (个数记为  $n_k$ )。树  $T$  中的结点总数  $n = n_0 + n_k = n_0 + m$ 。树中所含的边数  $e = n - 1$ , 这些边均为  $m$  个度为  $k$  的结点发出的,即  $e = m \times k$ 。整理得:

$$n_0 + m = m \times k + 1, \text{ 故 } n_0 = (k - 1) \times m + 1. \text{ (3 分)}$$

- (2) 高度为  $h$  的正则  $k$  叉树  $T$  中,含最多结点的树形为: 除第  $h$  层外,第 1 到第  $h-1$

层的结点都是度为  $k$  的分支结点，而第  $h$  层均为叶结点，即树是“满”树。此时第  $j$  ( $1 \leq j \leq h$ ) 层结点数为  $k^{j-1}$ ，结点总数  $M_1$  为：

$$M_1 = \sum_{j=1}^h k^{j-1} = \frac{k^h - 1}{k - 1} \quad (3 \text{ 分})$$

含最少结点的正则  $k$  叉树的树形为：第 1 层只有根结点，第 2 到第  $h-1$  层仅含 1 个分支结点和  $k-1$  个叶结点，第  $h$  层有  $k$  个叶结点。即除根外第 2 到第  $h$  层中每层的结点数均为  $k$ ，故  $T$  中所含结点总数  $M_2$  为：

$$M_2 = 1 + (h-1) \times k \quad (2 \text{ 分})$$

#### 【评分说明】

- ① 参考答案仅给出一种推导过程，若考生采用其他推导方法且正确，同样给分。
- ② 若考生仅给出结果，但没有推导过程，则 (1)、(2) 的最高得分分别是 2 分和 3 分。若推导过程或答案不完全正确，酌情给分。

#### 43. 【答案要点】

##### (1) 算法的基本设计思想 (4 分)

由题意知，将最小的  $\lfloor n/2 \rfloor$  个元素放在  $A_1$  中，其余的元素放在  $A_2$  中，分组结果即可满足题目要求。仿照快速排序的思想，基于枢轴将  $n$  个整数划分为两个子集。根据划分后枢轴所处的位置  $i$  分别处理：

- ① 若  $i = \lfloor n/2 \rfloor$ ，则分组完成，算法结束；
- ② 若  $i < \lfloor n/2 \rfloor$ ，则枢轴及之前的所有元素均属于  $A_1$ ，继续对  $i$  之后的元素进行划分；
- ③ 若  $i > \lfloor n/2 \rfloor$ ，则枢轴及之后的所有元素均属于  $A_2$ ，继续对  $i$  之前的元素进行划分；

基于该设计思想实现的算法，毋须对全部元素进行全排序，其平均时间复杂度是  $O(n)$ ，空间复杂度是  $O(1)$ 。

##### (2) 算法实现 (9 分)

int setPartition( int a[ ], int n )

```
{ int pivotkey, low = 0, low0 = 0, high = n - 1, high0 = n - 1, flag = 1, k = n / 2, i;
 int s1 = 0, s2 = 0;
 while (flag)
 { pivotkey = a[low]; // 选择枢轴
 while (low < high) // 基于枢轴对数据进行划分
 { while (low < high && a[high] >= pivotkey) --high;
 if (low != high) a[low] = a[high];
 while (low < high && a[low] <= pivotkey) ++low;
```

```

 if (low != high) a[high] = a[low];
 } // end of while (low < high)
 a[low] = pivotkey;
 if (low == k - 1) // 如果枢轴是第 n/2 小元素，划分成功
 flag = 0;
 else // 否则继续划分
 {
 if (low < k - 1)
 {
 low0 = ++ low;
 high = high0;
 }
 else
 {
 high0 = -- high;
 low = low0;
 }
 }
}
for (i = 0; i < k; i++) s1 += a[i];
for (i = k; i < n; i++) s2 += a[i];
return s2 - s1;
}

```

#### 【(1) (2) 的评分说明】

- ① 本题目只需将最大的一半元素与最小的一半元素分组，不需要对所有元素进行全部排序。参考答案基于快速排序思想，采用非递归的方式实现。若考生设计的算法满足题目的功能要求且正确，则（1）、（2）根据所实现算法的平均时间复杂度给分，细则见下表。

| 时间复杂度          | 分数 | 说明                     |
|----------------|----|------------------------|
| $O(n)$         | 13 | 采用类似快速排序思想，没有对元素进行全排序。 |
| $O(n\log_2 n)$ | 11 |                        |
| $O(n^2)$       | 9  |                        |
| 其他             | 7  | 时间复杂度高于 $O(n^2)$ 的算法。  |

- ② 若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路，但在算法实现中能够表达出算法思想且正确的，可参照①的标准给分。
- ③ 若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。
- ④ 参考答案中只给出了使用 C 语言的版本，使用 C++ 语言的答案视同使用 C 语言。
- (3) 算法的平均时间复杂度和空间复杂度 (2 分)

本参考答案给出的算法平均时间复杂度是  $O(n)$ ，空间复杂度是  $O(1)$ 。

【评分说明】若考生所估计的平均时间复杂度和空间复杂度与考生所实现的算法一致，可各给 1 分。

