

6.队列操作的原则是

- A.先进先出
- B.后进先出
- C.先进后出
- D.只进不出

7. 一棵深度为 6 的满二叉树有

- A. 63 个结点
- B.64 个结点
- C.127 个结点
- D.128 个结点

8.在一棵度为 3 的树中，度为 3 的结点有 4 个，度为 2 的结点有 2 个，度为 1 的结点有 3 个，则度为 0 的结点有

- A.8 个
- B.10 个
- C.11 个
- D.12 个

9. 一棵二叉树 T，度为 2 的结点数为 20 个，则叶子结点数为

- A. 19 个
- B.20 个
- C.21 个
- D.22 个

10. 有 10 个叶结点的哈夫曼树中共有

- A. 10 个结点
- B.11 个结点
- C.19 个结点
- D.21 个结点

11. 求图中两个结点之间的最短路径采用的算法是

- A. 广度优先搜索（BFS）算法
- B.克鲁斯卡尔（Kruskal）算法
- C.普里姆（Prim）算法
- D.迪杰斯特拉（Dijkstra）算法

12. 顺序查找算法的平均查找长度为

- A. $\log_2 n$
- B. $(n-1) / 2$
- C. $n/2$
- D. $(n+1) / 2$

13. 二叉排序树中，根的

- A. 左子树是二叉排序树、右子树不一定是二叉排序树
- B.左子树是二叉排序树、右子树也是二叉排序树
- C.左子树不一定是二叉排序树、右子树是二叉排序树
- D.左子树不一定是二叉排序树、右子树也不一定是二叉排序树

14. 冒泡排序的时间复杂度为

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

15. 关于稳定性的表述，正确的是

- A.稳定性是排序方法本身的特性，与数据无关
- B.稳定性不是排序方法本身的特性，与数据有关
- C.稳定性是排序方法本身的特性，与数据有关
- D.稳定性不是排序方法本身的特性，与数据无关

非选择题部分

注意事项:

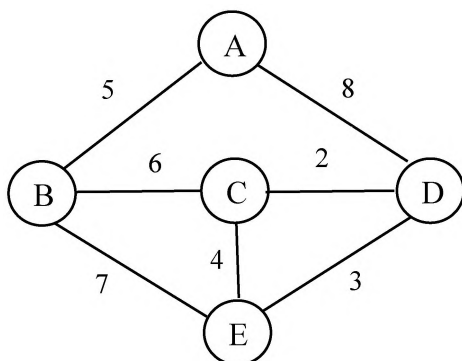
用黑色字迹的签字笔或钢笔将答案写在答题纸上，不能答在试题卷上。

二、填空题(本大题共 13 小题，每小题 2 分，共 26 分)

16. 数据中不可分割的最小标识单位是_____。
17. 双向循环链表中，在 p 所指结点的后面插入一个新结点*t，需要修改四个指针，分别为：t->prior=p; _____; p->next->prior=t; p->next=t;。
18. 在带有头结点的循环链表中，头指针为 head，判断指针 p 所指结点为首结点的条件是_____。
19. 元素的进栈次序为 1, 2, 3, ..., n，出栈的第一个元素是 n，则第 k 个出栈的元素是_____。
20. 在栈结构中，允许插入和删除的一端称为_____。
21. 100 个结点的二叉树采用三叉链表存储时，空指针域 NULL 有_____个。
22. 某二叉树的先序遍历序列为 ABKLMNO，中序遍历序列为 BLKANMO，则该二叉树中结点 A 的右孩子为结点_____。
23. 一个二叉树的最少结点个数为_____。
24. 图中第一个顶点和最后一个顶点相同的路径称为回路。除第一个顶点和最后一个顶点相同外，其余顶点不重复的回路，称为_____。
25. 设查找表有 n 个数据元素，则二分查找算法的平均查找长度为_____。
26. 用键值通过散列函数获取存储位置的这种存储方式构造的存储结构称为_____。
27. 若在线性表中采用二分查找法查找元素，则该线性表必须按值有序，并且采用_____存储结构。
28. 堆分为最小堆和最大堆，若键值序列 $\{k_1, k_2, \dots, k_n\}$ ，满足 $\begin{cases} k_i \geq k_{2i} \\ k_i \geq k_{2i+1} \end{cases} (i = 1, 2, \dots, \lfloor \frac{n}{2} \rfloor)$ ，则这 n 个键值序列 $\{k_1, k_2, \dots, k_n\}$ 是_____。

三、应用题(本大题共 5 小题，每小题 6 分，共 30 分)

29. 设一个链栈的输入序列为 X, Y, Z，试写出出栈的所有可能的输出序列及其操作步骤。
30. 设二叉树的先序遍历序列为 DCBAHEIFG，中序遍历序列为 ABCHDIEFG，试画出该二叉树并写出后序遍历序列。
31. 已知连通带权图如题 31 图所示，试利用普里姆 (Prim) 算法，从顶点 A 出发，构造它的最小生成树，画出构造过程。



题 31 图

32. 给定表 (28, 15, 55, 3, 71, 75, 10, 22, 56), 试按元素在表中的顺序将它们依次插入一棵初始时空的二叉排序树, 画出插入完成后的二叉排序树。

33. 应用直接选择排序算法, 对初始关键字序列为 48, 35, 61, 98, 82, 18, 29, 48 的记录进行从小到大排序, 写出排序过程和结果。

四、算法设计题(本大题共 2 小题, 每小题 7 分, 共 14 分)

34. 单链表的结点结构定义如下:

```
typedef struct node
{
    int data;
    struct node *next;
} Node, *LinkList;
```

试编写在带头结点的单链表 head 中查找第 1 个元素值小于 x 的结点的实现算法 Node *GetLinklist(LinkList head, int x), 若找到, 则返回指向该结点的指针, 否则返回 NULL。

35. 假设树采用孩子兄弟链表表示法, 其结构定义如下:

```
typedef struct tnode
{
    DataType data;
    struct tnode *son, *brother;
} *Tree;
```

试编写算法 void leveltree(Tree root) 实现树的按层次遍历。

2013 年 10 月高等教育自学考试全国统一命题考试

数据结构导论试题答案及评分参考

(课程代码 02142)

一、单项选择题(本大题共 15 小题,每小题 2 分,共 30 分)

- | | | | | |
|-------|-------|-------|-------|-------|
| 1. D | 2. D | 3. C | 4. A | 5. B |
| 6. A | 7. A | 8. C | 9. C | 10. C |
| 11. D | 12. D | 13. B | 14. C | 15. A |

二、填空题(本大题共 13 小题,每小题 2 分,共 26 分)

16. 数据项
17. $t \rightarrow next = p \rightarrow next$
18. $p == head \rightarrow next$
19. $n - k + 1$
20. 栈顶
21. 102
22. M
23. 0
24. 简单回路(或简单环)
25. $\frac{n+1}{n} \log_2(n+1) - 1$
26. 散列表
27. 顺序
28. 最大堆

三、应用题(本大题共 5 小题,每小题 6 分,共 30 分)

29. 共有五种可能的输出序列:

输出 XYZ, X 进, X 出, Y 进, Y 出, Z 进, Z 出 (1 分)

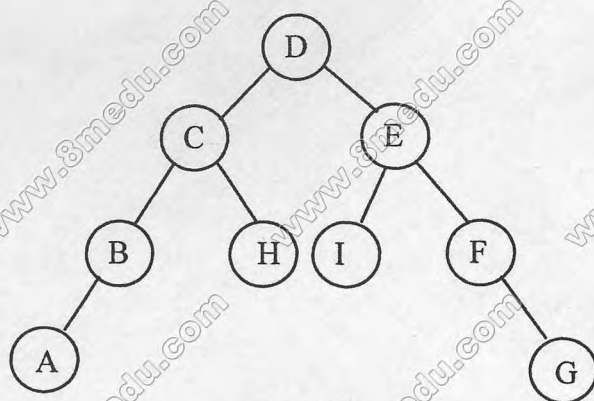
输出 YZX, X 进, Y 进, Y 出, Z 进, Z 出, X 出 (1 分)

输出 YXZ, X 进, Y 进, Y 出, X 出, Z 进, Z 出 (1 分)

输出 XZY, X 进, X 出, Y 进, Z 进, Z 出, Y 出 (1 分)

输出 ZYX, X 进, Y 进, Z 进, Z 出, Y 出, X 出 (2 分)

不能输出 ZXY。



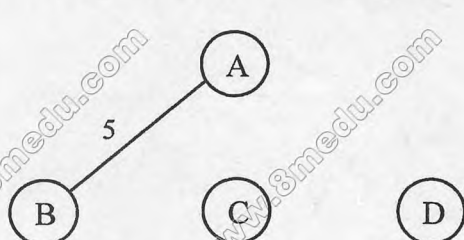
答 30 图

(左边 2 分, 右边 2 分)

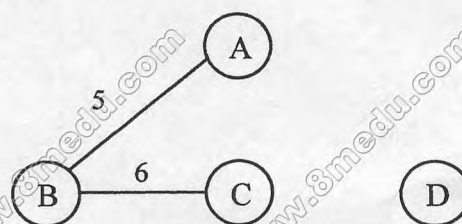
后序遍历序列为: ABHCIGFED

(2 分)

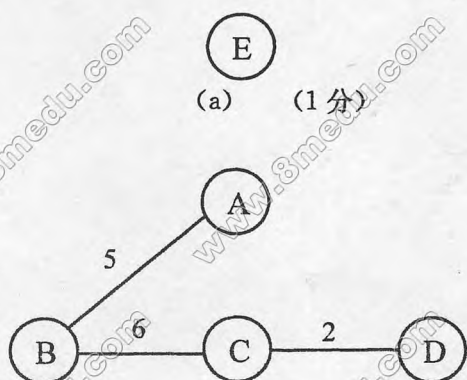
31. 最小生成树的构造过程如答 31 图(a)(b)(c)(d)所示。



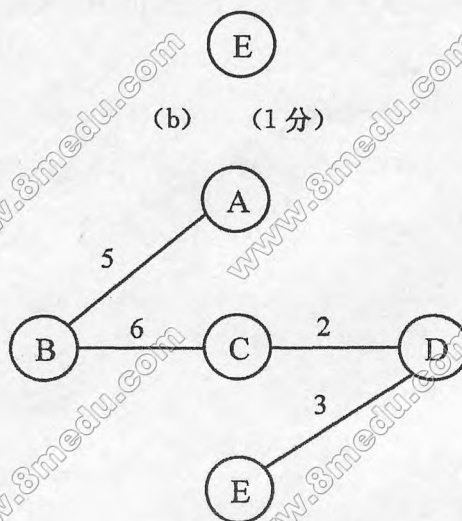
(a) (1 分)



(b) (1 分)

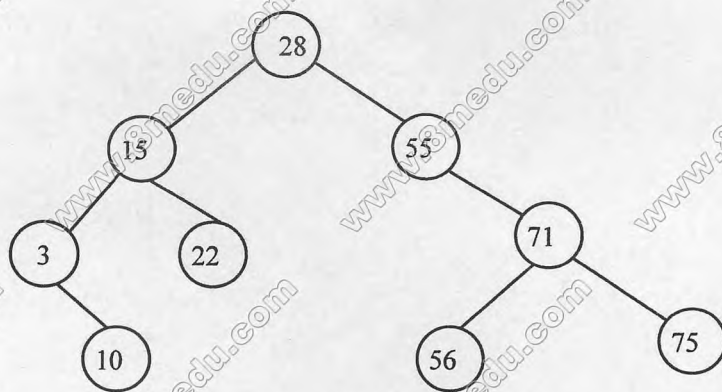


(c) (2 分)



(d) (2 分)

答 31 图



答 32 图

(左边 3 分, 右边 3 分)

33. 初始关键字: 48, 35, 61, 98, 82, 18, 29, 48

第一趟排序后: 18, 35, 61, 98, 82, 48, 29, 48

第二趟排序后: 18, 29, 61, 98, 82, 48, 35, 48 (1 分)

第三趟排序后: 18, 29, 35, 98, 82, 48, 61, 48 (1 分)

第四趟排序后: 18, 29, 35, 48, 82, 98, 61, 48 (1 分)

第五趟排序后: 18, 29, 35, 48, 48, 98, 61, 82 (1 分)

第六趟排序后: 18, 29, 35, 48, 48, 61, 98, 82 (1 分)

第七趟排序后: 18, 29, 35, 48, 48, 61, 82, 98 (1 分)

排序结果: 18, 29, 35, 48, 48, 61, 82, 98

四、算法设计题 (本大题共 2 小题, 每小题 7 分, 共 14 分)

34. Node * GetLinklist(LinkList head, int x)

```

{ Node * p;                                     (1 分)
  p = head->next;                                (2 分)
  while((p->data >= x) && (p! = NULL))           (2 分)
  {
    p = p->next;                                  (1 分)
  }
  return(p);                                     (1 分)
}

```

35. void leveltree(Tree root)

```

{ InitQueue(Q);                                //初始化队列 Q (1 分)
  if (root! = NULL)
  { EnQueue(Q, root);                          //根入队 (1 分)
  }
}

```

```
while (! EmptyQueue(Q)) //队列非空 (1 分)
```

```
{ p=GetHead(Q); //取队头元素
```

```
OutQueue(Q); //出队 (1 分)
```

```
printf(p->data);
```

```
p=p->son; (1 分)
```

```
while (p!=NULL)
```

```
{ EnQueue (Q,p); //p 入队 (1 分)
```

```
p=p->brother; (1 分)
```

```
}
```

```
}
```

```
}
```

```
}
```