

# 同济大学 2022—2023 学年期末

## 《数据结构》考试试卷（A 卷）

考试范围：《数据结构》；满分：100 分；考试时间：120 分钟

院/系：\_\_\_\_\_专业：\_\_\_\_\_姓名：\_\_\_\_\_考号：\_\_\_\_\_

题号	一	二	三	四	五	总分
得分						

注意事项：

1. 答题前填写好自己的姓名、专业、考号等信息
2. 本试题所有答案，应按试题顺序写在答题纸上，不必抄题，写清题号。写在试卷上不得分。

### 第 I 卷（选择题）

评卷人	得分

一、单项选择题：1~15 小题。下列每题给出的选项中，只有一个选项是符合题目要求的。

1. 具有 5 个叶子结点的二叉树中，度为 2 的结点的个数为（ ）  
A. 4  
B. 6  
C. 5  
D. 不确定
2. 设栈 S 和队列 Q 的初始状态为空，元素 e1, e2, e3, e4, e5 和 e6 依次进入栈 S 和进入队列 Q，若 6 个元素出队的序列是 e2, e4, e3, e6, e5, e1。则栈 S 的容量至少应该是（ ）  
A. 6  
B. 4  
C. 3  
D. 2
3. 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用（ ）存储方式最节省时间。  
A. 顺序表  
B. 双链表  
C. 带头结点的双循环链表  
D. 单循环链表
4. 有六个元素 6, 5, 4, 3, 2, 1 的顺序进栈，问下列哪一个不是合法的出栈序列？

( )

- A. 5 43 612
- B. 4 5312 6
- C. 3 4 6 521
- D. 23 41 5 6

5. 线性表是具有  $n$  个 ( ) 的有限序列 ( $n>0$ )

- A. 表元素
- B. 字符
- C. 数据元素
- D. 数据项

6. 对线性表进行二分查找时, 要求线性表必须 ( )

- A. 以顺序方式存储
- B. 以顺序方式存储, 且数据元素有序
- C. 以链接方式存储
- D. 以链接方式存储, 且数据元素有序

7. 散列函数有一个共同的性质, 即函数值应当以 ( ) 取其值域的每个值。

- A. 最大概率
- B. 最小概率
- C. 平均概率
- D. 同等概率

8. 判定一个队列  $Q$  (最多有  $n$  个元素) 为满的条件是 ( )。

- A.  $Q \rightarrow \text{rear} - Q \rightarrow \text{front} = n$
- B.  $Q \rightarrow \text{rear} - Q \rightarrow \text{front} + 1 = n$
- C.  $Q \rightarrow \text{rear} = Q \rightarrow \text{front}$
- D.  $Q \rightarrow \text{rear} + 1 = Q \rightarrow \text{front}$

9. 快速排序最易发挥其长处的情況是 ( )

- A. 被排序的数据中含有多个相同排序码
- B. 被排序的数据已基本有序
- C. 被排序的数据完全无序
- D. 被排序的数据中的最大值和最小值相差悬殊

10. 假设变址寄存器  $R$  的内容为  $1000H$ , 指令中的形式地址为  $2000H$ ; 地址  $1000H$  中的内容为  $2000H$ , 地址  $2000H$  中的内容为  $3000H$ , 地址  $3000H$  中的内容为  $4000H$ , 则变址寻方式下访问到的操作数是 ( )

- A.  $1000H$
- B.  $2000H$
- C.  $3000H$
- D.  $4000H$

11. 在下面的程序段中, 对  $x$  的赋值语句的频度为 ( )。

```
for (i=0; i<n; i++)  
for (j=0; j<n; j++)
```

$x=x+1;$

A.  $2n$

B.  $n$

C.  $n^2$

D.  $\log_2 n$

12. 在一棵具有 15 个关键字的 4 阶 B 树中，含关键字的结点数最多是 ( )

A. 5

B. 6

C. 10

D. 15

13. 外排序是指 ( )

A. 在外存上进行的排序方法。

B. 不需要使用内存的排序方法。

C. 数据量大，需要人工干预的排序方法。

D. 排序前后数据在外存，排序时数据调入内存的排序方法。

14. 有  $n$  个记录的文件，若关键字位数为  $d$ ，基数为  $r$ ，则基数排序共需进行 ( ) 遍分配与收集。

A.  $n$

B.  $r$

C.  $d$

D.  $d+r$

15. 通过 POP3 协议接收邮件时，使用的传输层服务类型是 ( )

A. 无连接不可靠的数据传输服务

B. 无连接可靠的数据传输服务

C. 有连接不可靠的数据传输服务

D. 有连接可靠的数据传输服务

## 第 II 卷（非选择题）

评卷人	得分

二、填空题：16~21 小题。请将答案写在答题纸指定位置上。

16. 组成串的数据元素只能是\_\_\_\_\_。

17. 数据的物理结构，指数据元素在\_\_\_\_\_中的标识（映象），也即\_\_\_\_\_。

18. 设  $n_0$  为哈夫曼树的叶结点数，则该哈夫曼树共有\_\_\_\_\_个结点。

19. 如果含  $n$  个顶点的图形形成一个环，则它有\_\_\_\_\_棵生成树。

20. 一个深度为  $k$  的，具有最少结点数的完全二叉树按层次，（同层次从左到右）用自然

数依此对结点编号，则编号最小的叶子的序号是\_\_\_\_\_；编号是 i 的结点所在的层次号是\_\_\_\_\_（根所在的层次号规定为 1 层）。

21. N 个顶点的连通图用邻接矩阵表示时，该矩阵至少有\_\_\_\_\_个非零元素。

评卷人	得分

### 三、判断题：22~29 小题。请将答案写在答题纸指定位置上。

22. 对长度为无穷大的广义表，由于存储空间的限制，不能在计算机中实现。（    ）
23. 即使对不含相同元素的同一输入序列进行两组不同的合法的入栈和出栈组合操作，所得的输出序列也一定相同。（    ）
24. 任何一个递归过程都可以转换成非递归过程。（    ）
25. 栈和队列都是限制存取点的线性结构。（    ）
26. 如果数据元素保持有序，则查找时就可以采用折半查找方法。（    ）
27. 二叉树是一棵无序树。（    ）
28. 内部排序是指排序过程在内存中进行的排序。（    ）
29. 若从二叉树的任一结点出发，到根的路径上所经过的结点序列按其关键字有序，则该二叉树一定是哈夫曼树。（    ）

评卷人	得分

### 四、程序填空：30~31 小题。请将答案写在答题纸指定位置上。

30. 根据需要，用适当的语句填入下面算法的\_\_\_\_\_中：

问题：设有 n 件物品，重量分别为 w1, w2, w3, ..., wn 和一个能装载总重量为 T 的背包。能否从 n 件物品中选择若干件恰好使它们的重量之和等于 T。若能，则背包问题有解，否则无解。解此问题的算法如下：

```

FUNCTION kanp_stack ( VAR stack , w:ARRAY[1..n] OF real ; VAR top:integer ;
T:real ) :boolean;
w[1: n] 存放 n 件物品的重量，依次从中取出物品放入背包中，检查背包重量，若不超过
T，则装入，否则弃之，取下一个物品试之。若有解则返回函数值 true，否则返回 false}
BEGIN
top:=0; i:=1; i 指示待选物品}
WHILE (1) _____ AND (2) _____ DO
[IF (3) _____ OR (4) _____ AND (i<n)
THEN [top := (5) _____ ; stack[top] :=i; 第 i 件物品装入背包}
T:=T-w[i]];
IF T=0 THEN RETURN ( (6) _____ ) 背包问题有解}
ELSE [IF (i=n) AND (top>0)
THEN [i:= (7) _____ ; 取出栈顶物品}
top:= (8) _____ ; T:= (9) _____ ]; 恢复 T 值}

```

```

i:=i+1    准备挑选下一件物品}
];
];
RETURN ( (10) _____ ) 背包无解}
END;

```

31. 下面程序段是逆转单向循环链表的方法，p0 是原链表头指针，逆转后链表头指针仍为 p0。

（可以根据需要增加标识符）

```

p:= p0; q0:=NIL;
WHILE (1) _____ DO
BEGIN (2) _____; (3) _____; (4) _____; (5) _____ END;
p^.next:= q0;  p0 ^.next:=p;  p0:=p;

```

评卷人	得分

**五、算法设计题：32~34 小题。请将答案写在答题纸指定位置上。**

32. 请用流程图或类高级语言（pascal 或 c）表示算法。已知有向图有 n 个顶点，请写算法，根据用户输入的偶对建立该有向图的邻接表。即接受用户输入的<vi, vj>（以其中之一为 0 标志结束），对于每条这样的边，申请一个结点，并插入到的单链表中，如此反复，直到将图中所有边处理完毕。提示：先产生邻接表的 n 个头结点（其结点数值域从 1 到 n）

33. 要求二叉树按二叉链表形式存储，

（1）写一个建立二叉树的算法。（2）写一个判别给定的二叉树是否是完全二叉树的算法。

完全二叉树定义为：深度为 K，具有 N 个结点的二叉树的每个结点都与深度为 K 的满二叉树中编号从 1 至 N 的结点一一对应。此题以此定义为准。

34. 设计算法以求解从集合 1..n} 中选取 k (k≤n) 个元素的所有组合。例如，从集合 1..4} 中选取 2 个元素的所有组合的输出结果为：1 2, 1 3, 1 4, 2 3, 2 4, 3 4。

【标准答案】

第 I 卷（选择题）

一、单项选择题：1~15 小题。下列每题给出的选项中，只有一个选项是符合题目要求的。

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

第 II 卷（非选择题）

二、填空题：16~21 小题。请将答案写在答题纸指定位置上。

16. 字符
17. 计算机；存储结构
18.  $2^{n0}-1$
19.  $n$
20.  $2^{k-2+1} \cdot \log_2 i + 1$
21.  $2(N-1)$

三、判断题：22~29 小题。请将答案写在答题纸指定位置上。

22.  $\sqrt{}$
23.  $\times$
24.  $\sqrt{}$
25.  $\sqrt{}$
26.  $\times$
27.  $\times$
28.  $\sqrt{}$

29. ×

四、程序填空：30~31 小题。请将答案写在答题纸指定位置上。

30. (1)  $T > 0$  (2)  $i < n$  (3)  $T > 0$  (4)  $top < n$  (5)  $top + 1$  (6) true (7)  $i - 1$  (8)  $top - 1$  (9)  $T + w[i]$  (10) false

31. (1)  $p \rightarrow next \neq p0$  (2)  $r := p \rightarrow next$  (3)  $p \rightarrow next := q0$ ;  
(4)  $q0 := p$ ; (5)  $p := r$

五、算法设计题：32~34 小题。请将答案写在答题纸指定位置上。

32. void CreatAdjList (AdjList g)

//建立有向图的邻接表存储结构

```
int n;
scanf ("%d",&n) ;
for (i=1;i<=n;j++)
scanf (&g[i].vertex) ; g[i].firstarc=null;}//输入顶点信息
scanf (&v1,&v2) ;
while (v1 && v2) //题目要求两顶点之一为 0 表示结束
i=GraphLocateVertex (g2,v1) ;
p= (ArcNode*) malloc (sizeof (ArcNode)) ;
p->adjvex=j; p->next=g[i].firstarc; g[i].firstarc=p;
scanf (&v1,&v2) ;
} }
```

33. BiTree Creat ( ) //建立二叉树的二叉链表形式的存储结构

ElemType x; BiTree bt;

scanf ("%d",&x) ; //本题假定结点数据域为整型

if (x==0) bt=null;

else if (x>0)

bt= (BiNode \*) malloc (sizeof (BiNode)) ;

bt->data=x; bt->lchild=creat ( ) ; bt->rchild=creat ( ) ;

}

else error ("输入错误") ;

return (bt) ;

//结束 BiTree

int JudgeComplete (BiTree bt) //判断二叉树是否是完全二叉树,如是, 返回 1, 否则, 返回 0

int tag=0; BiTree p=bt, Q[]; // Q 是队列, 元素是二叉树结点指针, 容量足够大

if (p==null) return (1) ;

QueueInit (Q) ; QueueIn (Q,p) ; //初始化队列, 根结点指针入队

while (!QueueEmpty (Q))

p=QueueOut (Q) ; //出队

if (p->lchild && !tag) QueueIn (Q,p->lchild) ; //左子女入队

else if (p->lchild) return 0; //前边已有结点为空, 本结点不空

```

else tag=1;           //首次出现结点为空
if (p->rchild && !tag) QueueIn (Q,p->rchild); //右子女入队
else if (p->rchild) return 0; else tag=1;
} //while
return 1; } //JudgeComplete
34. int A[],n; //设集合已存于数组 A 中。
void comb (int P[],int i,int k)
//从集合 (1..n) 中选取 k ( $k \leq n$ ) 个元素的所有组合
if (k==0) printf (P) ;
else if (k<=n) P[i]=A[i]; comb (P,i+1,k-1) ; comb (P,i+1,k) ; }
} //算法结束

```