



# 基础提高练习题

北街学长倾力之作

作者：北街

时间：2022/12/31

版本：1.0

## 目录

## 第一章 数据结构与算法

1. 设  $n$  是非负整数，程序段的时间复杂度是？

```
x = 0;
while(n >= (x+1)*(x+1))
    x = x + 1;
```

A.  $O(\log n)$    B.  $O(\sqrt{n})$    C.  $O(n)$    D.  $O(n^2)$

答案: **B.  $O(\sqrt{n})$**

解析: 该程序段的核心是一个 while 循环，循环条件为  $n \geq (x+1)^2$ 。在每次循环中， $x$  的值增加 1。循环的终止条件是  $x+1 > \sqrt{n}$ ，即  $x$  的最大值接近于  $\sqrt{n}$ 。

因此，循环的次数与  $\sqrt{n}$  成正比，时间复杂度为  $O(\sqrt{n})$ 。

- A.  $O(\log n)$ : 错误，循环次数与  $\log n$  无关。
- B.  $O(\sqrt{n})$ : 正确，循环次数与  $\sqrt{n}$  成正比。
- C.  $O(n)$ : 错误，循环次数远小于  $n$ 。
- D.  $O(n^2)$ : 错误，循环次数远小于  $n^2$ 。

2. 以下函数的时间复杂度是？

```
int func(int n) {
    int i = 0, sum = 0;
    while(sum < n)
        sum += ++i;
    return i;
}
```

A.  $O(\log n)$    B.  $O(\sqrt{n})$    C.  $O(n)$    D.  $O(n \log n)$

答案: **B.  $O(\sqrt{n})$**

解析: 该函数中，每次循环  $i$  的值增加 1， $sum$  的值增加  $i+1$ 。设执行了  $k$  次循环，则最终有:  $sum = 1 + 2 + 3 + \dots + k = \frac{k(k+1)}{2}$

当循环结束时，有  $sum \geq n$ ，即  $\frac{k(k+1)}{2} \geq n$ ，解得  $k \approx \sqrt{2n}$ 。

因此，循环的执行次数与  $\sqrt{n}$  成正比，时间复杂度为  $O(\sqrt{n})$ 。

- A.  $O(\log n)$ : 错误，循环次数与  $\log n$  无关。
- B.  $O(\sqrt{n})$ : 正确，循环次数与  $\sqrt{n}$  成正比。
- C.  $O(n)$ : 错误，循环次数小于  $n$ 。
- D.  $O(n \log n)$ : 错误，循环次数远小于  $n \log n$ 。

3. 程序片段的时间复杂度为：

```
x = 2;
while(x < n/2)
    x = 2 * x;
```

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

答案: **A.  $O(\log_2 n)$**

解析：该程序段中， $x$  的初始值为 2，每次循环  $x$  变为原来的两倍，即  $x$  的值依次为：2, 4, 8, 16, ...,  $2^k$ 。  
循环终止的条件是  $x \geq n/2$ ，即  $2^k \geq n/2$ ，解得  $k \geq \log_2(n/2) = \log_2 n - 1$ 。

因此，循环的执行次数与  $\log_2 n$  成正比，时间复杂度为  $O(\log_2 n)$ 。

- A.  $O(\log_2 n)$ ：正确，循环次数与  $\log_2 n$  成正比。
- B.  $O(n)$ ：错误，循环次数远小于  $n$ 。
- C.  $O(n \log_2 n)$ ：错误，循环次数远小于  $n \log_2 n$ 。
- D.  $O(n^2)$ ：错误，循环次数远小于  $n^2$ 。

4. 阶乘算法时间复杂度为？

```
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.  $O(n^2)$

答案：B.  $O(n)$

解析：该函数是计算阶乘的递归算法。函数的递归调用次数与  $n$  的值直接相关。具体来说，对于输入  $n$ ，函数将递归调用自身  $n-1$  次，直到基本情况  $n \leq 1$ 。

因此，时间复杂度为  $O(n)$ 。

- A.  $O(\log_2 n)$ ：错误，递归调用次数不是对数级的。
- B.  $O(n)$ ：正确，递归调用次数与  $n$  成正比。
- C.  $O(n \log_2 n)$ ：错误，递归调用次数小于  $n \log_2 n$ 。
- D.  $O(n^2)$ ：错误，递归调用次数小于  $n^2$ 。

5. 已知两个长度分别为  $m$  和  $n$  的升序链表，合并这两个升序链表为降序链表的最坏时间复杂度为？

A.  $O(n)$    B.  $O(mn)$    C.  $O(\min(m, n))$    D.  $O(\max(m, n))$

答案：D.  $O(\max(m, n))$

解析：合并两个升序链表为一个降序链表的过程需要两个步骤：1. 合并两个升序链表为一个升序链表，时间复杂度为  $O(m+n)$ ，因为每个元素最多比较一次。2. 将合并后的升序链表逆序，时间复杂度为  $O(m+n)$ 。因此，总的时间复杂度为  $O(m+n)$ ，即  $O(\max(m, n))$ （当  $m$  和  $n$  中较大的一个远大于另一个时）。

- A.  $O(n)$ ：部分正确，但不完整，因为没有考虑  $m$  的情况。
- B.  $O(mn)$ ：错误，合并操作不需要  $m \times n$  次比较。
- C.  $O(\min(m, n))$ ：错误，即使较短的链表处理完，还需要处理较长的链表。
- D.  $O(\max(m, n))$ ：正确，等价于  $O(m+n)$ 。

6. 以下程序段时间复杂度为：

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

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

答案：C.  $O(n \log n)$

解析：该程序段包含嵌套循环。外层循环中， $k$  的值以 2 为底的指数增长，即  $k$  的值依次为：1, 2, 4, 8, ...,  $2^{\lfloor \log_2 n \rfloor}$ 。外层循环的执行次数为  $\lfloor \log_2 n \rfloor + 1$ ，即  $O(\log n)$ 。

内层循环的执行次数固定为  $n$ 。

因此，总的执行次数为外层循环次数乘以内层循环次数，即  $O(\log n) \times O(n) = O(n \log n)$ 。



- A.  $O(\log n)$ : 错误, 没有考虑内层循环。
- B.  $O(n)$ : 错误, 总执行次数大于  $n$ 。
- C.  $O(n \log n)$ : 正确, 是外层循环次数与内层循环次数的乘积。
- D.  $O(n^2)$ : 错误, 总执行次数小于  $n^2$ 。

7. 数据的最小单位是?

A. 数据项 B. 字节 C. 数据元素 D. 结点

答案: **A. 数据项**

解析: 在数据结构中, 数据的层次关系如下: - 数据项: 是数据的最小单位, 是不可分割的原子单位, 是针对具体问题中某个数据对象的某个属性的取值。- 数据元素: 是数据的基本单位, 由若干个数据项组成。- 数据对象: 是具有相同性质的数据元素的集合。- 数据结构: 是相互之间存在一种或多种特定关系的数据元素的集合。

因此, 数据的最小单位是数据项。

- A. 数据项: 正确, 是数据的最小单位。
- B. 字节: 错误, 字节是计算机存储的基本单位, 不是数据结构中的概念。
- C. 数据元素: 错误, 数据元素是数据的基本单位, 由若干个数据项组成。
- D. 结点: 错误, 结点是存储数据元素的物理单位。

8. 数据的基本单位是?

A. 数据项 B. 数据类型 C. 数据元素 D. 数据变量

答案: **C. 数据元素**

解析: 在数据结构中: - 数据元素是数据的基本单位, 是数据结构中讨论的基本对象。例如, 一个学生记录就是一个数据元素。- 数据项是数据的最小单位, 是组成数据元素的不可分割的最小单位。例如, 学生记录中的学号、姓名等都是数据项。

因此, 数据的基本单位是数据元素。

- A. 数据项: 错误, 数据项是数据的最小单位, 不是基本单位。
- B. 数据类型: 错误, 数据类型描述了数据的取值范围和对数据可执行的操作。
- C. 数据元素: 正确, 是数据的基本单位。
- D. 数据变量: 错误, 数据变量是程序设计中的概念, 不是数据结构的基本概念。

9. 数据对象是指?

A. 描述客观事物且由计算机处理的数值、字符等符号的总称 B. 数据的基本单位 C. 同类数据元素集合 D. 相互之间存在一种或多种特定关系的数据元素的集合

答案: **C. 同类数据元素集合**

解析: 在数据结构中: - 数据是描述客观事物的符号, 是计算机可以操作的对象。- 数据元素是数据的基本单位。- 数据对象是具有相同性质的数据元素的集合, 即同类数据元素的集合。- 数据结构是相互之间存在一种或多种特定关系的数据元素的集合。

因此, 数据对象是指同类数据元素的集合。

- A. 描述客观事物且由计算机处理的数值、字符等符号的总称: 错误, 这是对数据的描述。
- B. 数据的基本单位: 错误, 这是对数据元素的描述。
- C. 同类数据元素集合: 正确, 这是数据对象的定义。
- D. 相互之间存在一种或多种特定关系的数据元素的集合: 错误, 这是对数据结构的描述。

10. 以下说法正确的是?

A. 数据元素是最小单位 B. 数据项是基本单位 C. 数据结构是结构化的数据元素集合 D. 数据结构是结构化的数据项集合

答案: **C. 数据结构是结构化的数据元素集合**

解析: 根据数据结构的基本概念: - 数据项是数据的最小单位, 不可分割。- 数据元素是数据的基本单位, 由若干个数据项组成。- 数据结构是相互之间存在一种或多种特定关系 (即结构化) 的数据元素的集合。

因此，正确的说法是“数据结构是结构化的数据元素集合”。

- A. 数据元素是最小单位：错误，数据项才是最小单位。
- B. 数据项是基本单位：错误，数据元素才是基本单位。
- C. 数据结构是结构化的数据元素集合：正确。
- D. 数据结构是结构化的数据项集合：错误，数据结构是数据元素（不是数据项）的集合。

11. 数据结构研究内容包括？

A. 数据组织 B. 数据存储 C. 运算实现 D. 描述语言

答案：A. 数据组织, B. 数据存储, C. 运算实现

解析：数据结构的研究内容主要包括三个方面：1. 数据的逻辑结构：研究数据元素之间的逻辑关系，即数据组织方式。2. 数据的存储结构：研究数据在计算机中的存储表示。3. 数据的操作：研究对数据进行各种运算的实现算法。

描述语言不是数据结构的主要研究内容，而是表达数据结构的工具。

- A. 数据组织：正确，属于数据结构的研究内容之一。
- B. 数据存储：正确，属于数据结构的研究内容之一。
- C. 运算实现：正确，属于数据结构的研究内容之一。
- D. 描述语言：错误，不是数据结构的主要研究内容。

12. 定义 ADT 除数据对象与关系，还需说明？

A. 数据元素 B. 算法描述 C. 基本操作 D. 数据项

答案：C. 基本操作

解析：抽象数据类型（ADT, Abstract Data Type）是一个数学模型，它由三部分组成：1. 数据对象：即该数据类型中包含的数据元素的集合。2. 数据关系：即数据对象中各元素之间的关系。3. 基本操作：即对数据类型的各种操作。

因此，定义 ADT 除了需要说明数据对象与关系外，还需要说明基本操作。

- A. 数据元素：错误，数据元素是组成数据对象的基本单位，已包含在数据对象中。
- B. 算法描述：错误，算法描述是基本操作的具体实现方法，不是 ADT 定义的一部分。
- C. 基本操作：正确，是 ADT 定义的必要组成部分。
- D. 数据项：错误，数据项是数据元素的组成部分，不是 ADT 定义的必要组成部分。

13. 数据结构可逻辑划分为？

A. 动态/静态结构 B. 紧凑/非紧凑结构 C. 内部/外部结构 D. 线性/非线性结构

答案：D. 线性/非线性结构

解析：数据结构按逻辑关系可以划分为线性结构和非线性结构：- 线性结构：数据元素之间存在一对一的线性关系，如线性表、栈、队列、串等。- 非线性结构：数据元素之间存在一对多或多对多的关系，如树、图等。

而动态/静态、紧凑/非紧凑、内部/外部等划分方式不是针对数据的逻辑结构，而是针对数据的存储结构或其他特性的划分。

- A. 动态/静态结构：错误，这是按照存储空间的分配方式划分的。
- B. 紧凑/非紧凑结构：错误，这不是数据结构的标准划分方式。
- C. 内部/外部结构：错误，这是按照数据存储在内存还是外存划分的。
- D. 线性/非线性结构：正确，这是按照数据元素之间的逻辑关系划分的。

14. 以下结构属于逻辑结构的是？

A. 顺序表 B. 哈希表 C. 有序表 D. 单链表

答案：C. 有序表

解析：数据的逻辑结构是从问题出发，考虑问题所涉及的数据元素之间的逻辑关系，与数据的存储无关。逻辑结构主要分为线性结构和非线性结构。

- 顺序表：是线性表的一种存储结构，属于存储结构，不是逻辑结构。- 哈希表：是一种特殊的存储结构，

用于实现关联数组，不是逻辑结构。- 有序表：是一种逻辑结构，表示元素之间存在顺序关系的线性表。- 单链表：是线性表的一种存储结构，属于存储结构，不是逻辑结构。

- A. 顺序表：错误，属于存储结构。
- B. 哈希表：错误，属于存储结构。
- C. 有序表：正确，属于逻辑结构。
- D. 单链表：错误，属于存储结构。

15. 数据元素之间的逻辑关系被称为？

A. 数据的存储结构 B. 数据的基本操作 C. 程序的算法 D. 数据的逻辑结构

答案：D. 数据的逻辑结构

解析：数据结构包括三方面的内容：逻辑结构、存储结构和操作。

- 逻辑结构：描述数据元素之间的逻辑关系，与存储无关。- 存储结构：描述数据元素在计算机内存中的存储方式。- 操作：研究对数据进行各种运算的实现算法。

因此，数据元素之间的逻辑关系被称为数据的逻辑结构。

- A. 数据的存储结构：错误，存储结构描述的是数据元素的存储表示。
- B. 数据的基本操作：错误，基本操作描述的是数据结构支持的操作。
- C. 程序的算法：错误，算法是解决问题的步骤。
- D. 数据的逻辑结构：正确，逻辑结构描述的是数据元素之间的逻辑关系。

16. 以下与数据的存储结构无关的是？

A. 循环队列 B. 链表 C. 哈希表 D. 栈

答案：D. 栈

解析：存储结构是数据逻辑结构在计算机中的存储表示，主要包括顺序存储、链式存储、索引存储和散列存储等。

- 循环队列：是队列的一种顺序存储实现方式，属于存储结构。- 链表：是一种链式存储结构，属于存储结构。- 哈希表：是一种散列存储结构，属于存储结构。- 栈：是一种逻辑结构，是限定在一端进行插入和删除操作的线性表。栈有顺序栈和链栈两种存储实现方式，但栈本身是一种逻辑结构。

- A. 循环队列：错误，循环队列是队列的一种存储结构。
- B. 链表：错误，链表是一种存储结构。
- C. 哈希表：错误，哈希表是一种存储结构。
- D. 栈：正确，栈是一种逻辑结构，与存储结构无关。

17. 以下数据结构中，哪一个是线性结构？

A. 广义表 B. 二叉树 C. 稀疏矩阵 D. 串

答案：D. 串

解析：线性结构指的是数据元素之间存在一对一的线性关系的数据结构，即除了第一个和最后一个元素外，每个元素都有唯一的前驱和后继。

- 广义表：是一种非线性的数据结构，因为其元素可以是原子（基本元素）或者子表。- 二叉树：是一种非线性结构，每个结点最多有两个子结点。- 稀疏矩阵：是矩阵的一种特殊形式，是一种二维结构，属于非线性结构。- 串（字符串）：是由字符组成的一维线性序列，属于线性结构。

- A. 广义表：错误，属于非线性结构。
- B. 二叉树：错误，属于非线性结构。
- C. 稀疏矩阵：错误，属于非线性结构。
- D. 串：正确，属于线性结构。

18. 以下哪个数据结构不是多型数据类型？

A. 栈 B. 广义表 C. 有向图 D. 字符串

答案：D. 字符串

解析：多型数据类型指的是其数据元素的类型不唯一的数据类型。

- 栈：可以存储不同类型的数据元素，如整数、字符、结构体等，是多型数据类型。- 广义表：其元素可以是原子或者子表，是多型数据类型。- 有向图：图中的顶点可以存储不同类型的数据，是多型数据类型。- 字符串：其元素类型必须是字符，是单一类型的数据结构，不是多型数据类型。

- A. 栈：错误，栈可以存储不同类型的数据元素，是多型数据类型。
- B. 广义表：错误，广义表的元素可以是原子或子表，是多型数据类型。
- C. 有向图：错误，图中的顶点可以存储不同类型的数据，是多型数据类型。
- D. 字符串：正确，字符串只能存储字符类型的数据，不是多型数据类型。

19. 以下数据结构中，属于非线性结构的是？

A. 树 B. 字符串 C. 队列 D. 栈

答案：A. 树

解析：数据结构按照逻辑关系可以分为线性结构和非线性结构：- 线性结构：数据元素之间存在一对一的线性关系，如线性表、栈、队列、字符串等。- 非线性结构：数据元素之间存在一对多或多对多的关系，如树、图等。

- 树：每个结点可以有零个或多个子结点，属于非线性结构。- 字符串：是由字符组成的一维线性序列，属于线性结构。- 队列：是一种特殊的线性表，只允许在一端插入，另一端删除，属于线性结构。- 栈：是一种特殊的线性表，只允许在一端进行插入和删除操作，属于线性结构。

- A. 树：正确，树是典型的非线性结构。
- B. 字符串：错误，字符串是线性结构。
- C. 队列：错误，队列是线性结构。
- D. 栈：错误，栈是线性结构。

20. 下列数据中，属于非线性结构的是？

A. 栈 B. 队列 C. 完全二叉树 D. 堆

答案：C. 完全二叉树

解析：根据数据元素之间的逻辑关系，数据结构可以分为线性结构和非线性结构。

- 栈：是一种特殊的线性表，只允许在一端进行插入和删除操作，属于线性结构。- 队列：是一种特殊的线性表，只允许在一端插入，另一端删除，属于线性结构。- 完全二叉树：是一种特殊的二叉树，属于非线性结构。- 堆：通常指的是二叉堆，是一种特殊的完全二叉树，也是线性数据结构（数组）的逻辑解释。但从逻辑结构上看，堆是一种树状结构，属于非线性结构。

这道题的选项 C 和 D 都可以被视为非线性结构，但完全二叉树更明确地属于非线性结构。

- A. 栈：错误，栈是线性结构。
- B. 队列：错误，队列是线性结构。
- C. 完全二叉树：正确，完全二叉树是非线性结构。
- D. 堆：正确，堆是非线性结构，但通常堆的实现是使用数组。

21. 连续存储设计中，存储地址？

A. 一定连续 B. 一定不连续 C. 不一定连续 D. 部分连续，部分不连续

答案：A. 一定连续

解析：存储结构按照实现方式可以分为两大类：- 顺序存储结构：用一组地址连续的存储单元依次存储数据元素，相邻数据元素的存储位置也相邻。- 链式存储结构：用一组任意的存储单元存储数据元素，这组存储单元可以是连续的，也可以是不连续的。

连续存储设计即是顺序存储结构，其特点是存储单元的地址一定是连续的。

- A. 一定连续：正确，连续存储设计中数据元素的存储地址一定连续。
- B. 一定不连续：错误，连续存储设计的定义就是存储地址连续。
- C. 不一定连续：错误，连续存储设计中存储地址必须连续。
- D. 部分连续，部分不连续：错误，这种情况下不能称为连续存储设计。

22. 以下属于逻辑结构的是？



A. 顺序表 B. 哈希表 C. 有序表 D. 单链表

答案: **C. 有序表**

解析: 数据结构分为逻辑结构和存储结构: - 逻辑结构: 描述数据元素之间的逻辑关系, 与存储无关。- 存储结构: 描述数据元素在计算机内存中的存储方式。

- 顺序表: 是线性表的一种顺序存储实现, 属于存储结构。- 哈希表: 是一种散列存储结构, 属于存储结构。  
- 有序表: 是一种逻辑结构, 表示元素之间存在顺序关系的线性表。- 单链表: 是线性表的一种链式存储实现, 属于存储结构。

- A. 顺序表: 错误, 属于存储结构。
- B. 哈希表: 错误, 属于存储结构。
- C. 有序表: 正确, 属于逻辑结构。
- D. 单链表: 错误, 属于存储结构。

23. 算法的计算量的大小称为计算的?

A. 效率 B. 复杂性 C. 现实性 D. 难度

答案: **B. 复杂性**

解析: 算法的计算量是指算法执行过程中所需的计算操作次数, 它是算法复杂性的一个衡量标准。算法的复杂性一般分为时间复杂性和空间复杂性: - 时间复杂性: 度量算法执行时间随问题规模增大而增长的关系。- 空间复杂性: 度量算法占用存储空间随问题规模增大而增长的关系。

因此, 算法的计算量的大小称为计算的复杂性。

- A. 效率: 错误, 效率更多地是指算法的实际执行表现。
- B. 复杂性: 正确, 复杂性是对算法计算量大小的描述。
- C. 现实性: 错误, 现实性指的是算法是否切实可行。
- D. 难度: 错误, 难度更多地是指理解或实现算法的困难程度。

24. 算法的时间复杂度取决于?

A. 问题的规模 B. 待处理数据的初态 C. A 和 B

答案: **C. A 和 B**

解析: 算法的时间复杂度主要取决于两个因素: 1. 问题的规模: 一般来说, 问题规模越大, 算法执行的时间就越长。2. 待处理数据的初态: 对于同样规模的问题, 由于数据的初始状态不同, 算法的执行效率也会有所不同。例如, 对于排序算法, 如果数据已经基本有序, 某些算法的执行时间会大大降低。

因此, 算法的时间复杂度既取决于问题的规模, 也取决于待处理数据的初始状态。

- A. 问题的规模: 部分正确, 但不完整。
- B. 待处理数据的初态: 部分正确, 但不完整。
- C. A 和 B: 正确, 算法的时间复杂度取决于问题规模和数据初态。

25. 算法必须具备的特性?

1. A. 计算方法 B. 排序方法 C. 解决问题的步骤序列 D. 调度方法

2. A. 可执行性、可移植性、可扩充性 B. 可执行性、确定性、有穷性

C. 确定性、有穷性、稳定性 D. 易读性、稳定性、安全性

答案: 1. **C. 解决问题的步骤序列**, 2. **B. 可执行性、确定性、有穷性**

解析: 1. 算法的定义是解决问题的步骤序列, 是为了解决某一问题而采取的明确、有限的操作步骤。2. 算法必须具备的五个基本特性是: - 有穷性: 算法必须在有限步骤内结束。- 确定性: 算法的每一步骤必须有明确的定义, 不会产生二义性。- 可行性(可执行性): 算法的每一步操作必须是可行的, 即能够通过已经实现的基本运算执行有限次而实现。- 输入: 算法可以有零个或多个输入。- 输出: 算法必须有一个或多个输出。

其中, 可执行性、确定性和有穷性是最基本的三个特性。

- 1.C. 解决问题的步骤序列: 正确, 这是算法的定义。
- 2.B. 可执行性、确定性、有穷性: 正确, 这是算法必须具备的三个基本特性。

26. 一个算法应该是？

A. 程序 B. 问题求解步骤的描述 C. 要满足五个基本特性 D. A 和 C

答案：B. 问题求解步骤的描述

解析：算法是指解决问题的方法和步骤，是对解决问题的过程的描述。算法需要满足五个基本特性（有穷性、确定性、可行性、输入、输出），但算法本身是问题求解步骤的描述，而不是特性的集合。

程序是算法的一种具体实现，算法可以用自然语言、流程图或伪代码等形式表示，不一定是计算机程序。

- A. 程序：错误，程序是算法的一种具体实现，但算法不等同于程序。
- B. 问题求解步骤的描述：正确，这是算法的本质定义。
- C. 要满足五个基本特性：错误，这是对算法的要求，而不是算法本身的定义。
- D. A 和 C：错误，这是把程序和算法特性错误地混为一谈。

27. 错误说法是？

(1) 算法原地工作不需额外空间

(2)  $O(n)$  一定快于  $O(n^2)$

(3) 时间复杂度为最坏情况估算

(4) 编程语言越高级效率越低

A. (1) B. (1)(2) C. (1)(4) D. (3)

答案：C. (1)(4)

解析：分析每个说法：

(1) 算法原地工作不需额外空间：错误。原地工作算法指的是在算法执行过程中不需要额外的数据结构来存储中间结果，但并不意味着不需要任何额外空间。算法总是需要一些临时变量空间来存储中间计算结果，最少也需要  $O(1)$  的额外空间。

(2)  $O(n)$  一定快于  $O(n^2)$ ：正确，但有条件。从渐近意义上讲，当问题规模  $n$  足够大时， $O(n)$  的算法一定快于  $O(n^2)$  的算法。但对于特定的小规模问题或特定的输入， $O(n^2)$  的算法可能比  $O(n)$  的算法更快。

(3) 时间复杂度为最坏情况估算：正确。通常我们使用时间复杂度来估算算法的执行时间上限，即最坏情况下的执行时间。

(4) 编程语言越高级效率越低：错误。这是一个过于简化的说法。虽然高级语言可能引入一些额外开销，但现代编译器和优化技术可以显著减少这些差异。此外，使用高级语言可以更容易地实现复杂算法，有时甚至可以获得更高的效率。

- A. (1)：错误，(1) 确实是错误的，但 (4) 也是错误的。
- B. (1)(2)：错误，(2) 在渐近意义上是正确的。
- C. (1)(4)：正确，(1) 和 (4) 都是错误的说法。
- D. (3)：错误，(3) 是正确的说法。

28. 时间复杂度计算方法属于？

A. 事前统计 B. 事前估算 C. 事后统计 D. 事后分析

答案：B. 事前估算

解析：时间复杂度是在算法设计阶段对算法执行时间的一种理论估计，它是在算法实际执行前进行的分析。

- 事前统计：不适用于时间复杂度，因为时间复杂度不是通过统计得出的。

- 事前估算：正确，时间复杂度是对算法执行时间的事前理论估算。

- 事后统计：不适用于时间复杂度，这更像是性能测试的过程。

- 事后分析：不适用于时间复杂度，时间复杂度分析是在算法设计阶段进行的。

- A. 事前统计：错误，时间复杂度不是通过统计确定的。
- B. 事前估算：正确，时间复杂度是对算法执行时间的事前理论估算。
- C. 事后统计：错误，这涉及到实际执行后的数据收集。
- D. 事后分析：错误，时间复杂度分析是在实际执行前进行的。

29. 定义完整数据结构所需？

A. 数据元素 B. 数据对象 C. 数据关系 D. 抽象数据类型

答案: **D. 抽象数据类型**

解析: 定义一个完整的数据结构需要抽象数据类型 (ADT), 它包括以下三个方面: 1. 数据对象: 数据元素的集合。2. 数据关系: 数据元素之间的关系。3. 基本操作: 对数据结构的各种操作。

抽象数据类型是一个完整的概念, 包含了数据的定义和操作的完整描述。仅仅有数据元素、数据对象或数据关系是不足以定义一个完整的数据结构的。

- A. 数据元素: 错误, 数据元素只是数据结构的基本单位, 不足以定义完整数据结构。
- B. 数据对象: 错误, 数据对象虽然包含了数据元素集合, 但缺少关系和操作的描述。
- C. 数据关系: 错误, 数据关系只描述了元素之间的关系, 缺少对象和操作的描述。
- D. 抽象数据类型: 正确, ADT 包含了定义完整数据结构所需的所有要素。

30. 合法输入时算法应具备?

A. 可读性 B. 健壮性 C. 正确性 D. 有穷性

答案: **C. 正确性**

解析: 对于合法的输入, 算法最基本的要求是正确性, 即能够得到正确的结果。

- 可读性: 是指算法易于理解和修改, 与算法对合法输入的处理能力无关。- 健壮性: 是指算法对非法输入的处理能力, 而题目中明确说明了是合法输入。- 正确性: 是指算法对合法输入能够产生正确的输出结果。

- 有穷性: 是指算法必须在有限步骤内结束, 这是算法的基本特性之一, 但不是针对合法输入的特殊要求。

- A. 可读性: 错误, 可读性是算法实现的质量特性, 与处理合法输入无关。
- B. 健壮性: 错误, 健壮性是针对非法输入的处理能力。
- C. 正确性: 正确, 合法输入时算法首先应该保证输出正确结果。
- D. 有穷性: 错误, 有穷性是算法的基本特性, 不是针对合法输入的特殊要求。

31. 算法分析的目的?

A. 合理性分析 B. 输入输出研究 C. 效率分析与改进 D. 易懂性分析

答案: **C. 效率分析与改进**

解析: 算法分析是对算法的时间复杂度和空间复杂度进行评估, 以便比较不同算法的效率并寻找改进的方法。算法分析的主要目的是:

- 确定算法所需的计算资源量 (时间和空间)。- 比较不同算法的效率, 选择更优的算法。- 分析算法的瓶颈, 寻找改进的方向。

因此, 算法分析的主要目的是效率分析与改进。

- A. 合理性分析: 错误, 合理性分析更多地是针对算法的正确性和可行性。
- B. 输入输出研究: 错误, 输入输出研究是算法设计的一部分, 不是分析的主要目的。
- C. 效率分析与改进: 正确, 算法分析的主要目的是评估和改进算法的效率。
- D. 易懂性分析: 错误, 易懂性分析更多地关注算法的可读性和可维护性。

32. 好算法应具备的目标?

A. 可行性 B. 健壮性 C. 无二义性 D. 可读性好

答案: **A. 可行性, B. 健壮性, C. 无二义性, D. 可读性好**

解析: 一个好的算法应该具备以下特性:

- 可行性 (可执行性): 算法的每一步都必须是可行的, 能够通过已经实现的基本运算执行有限次而实现。

- 健壮性: 算法应能够对非法输入进行适当处理, 而不是产生错误的结果或者崩溃。- 无二义性 (确定性):

算法的每一步都应有明确的定义, 不会产生歧义。- 可读性好: 算法应该清晰明了, 便于理解和修改。

此外, 好的算法还应具备有穷性 (在有限步骤内结束)、正确性 (对合法输入能产生正确的输出) 和高效性 (时间和空间效率高) 等特性。

- A. 可行性: 正确, 好算法必须是可行的。
- B. 健壮性: 正确, 好算法应该具有健壮性。
- C. 无二义性: 正确, 好算法应该是确定的, 没有二义性。
- D. 可读性好: 正确, 好算法应该易于理解和维护。

33. 数据元素之间的关系称为?

A. 操作 B. 结构 C. 数据对象 D. 数据集合

答案: **B. 结构**

解析: 在数据结构中:

- 数据元素是数据的基本单位。- 数据元素之间的关系称为数据的结构, 也就是逻辑结构。- 数据对象是具有相同性质的数据元素的集合。- 数据集合是数据元素的集合, 强调的是元素的收集性而非它们之间的关系。- 操作是对数据进行的各种处理, 如增、删、改、查等。

- A. 操作: 错误, 操作是对数据进行的处理, 不是数据元素之间的关系。
- B. 结构: 正确, 数据元素之间的关系定义了数据的结构。
- C. 数据对象: 错误, 数据对象是数据元素的集合, 不是元素之间的关系。
- D. 数据集合: 错误, 数据集合强调的是元素的收集性, 不强调元素之间的关系。

34. 算法的特点 (多选)?

A. 有  $\geq 0$  个输入量 B. 健壮性 C. 正确性 D. 可行性

答案: **A. 有  $\geq 0$  个输入量, C. 正确性, D. 可行性**

解析: 算法的五个基本特性是:

1. 有穷性: 算法必须在有限步骤内结束。2. 确定性: 算法的每一步骤必须有明确的定义, 不会产生二义性。3. 可行性 (可执行性): 算法的每一步操作必须是可行的, 即能够通过已经实现的基本运算执行有限次而实现。4. 输入: 算法可以有零个或多个输入。5. 输出: 算法必须有一个或多个输出。

健壮性是算法的一个重要但非基本的特性, 它指的是算法对非法输入的处理能力。正确性是算法的一个本质特性, 指算法对合法输入能够产生正确的输出。

- A. 有  $\geq 0$  个输入量: 正确, 算法可以有零个或多个输入。
- B. 健壮性: 错误, 健壮性不是算法的基本特性。
- C. 正确性: 正确, 算法必须对合法输入产生正确的输出。
- D. 可行性: 正确, 算法的每一步都必须是可行的。

35. 以下程序的时间复杂度?

```
for (int i = 0; i < m; i++)
    for (int j = 0; j < n; j++)
        a[i][j] = 0;
```

A.  $O(n^2)$  B.  $O(mn)$  C.  $O(m^2)$  D.  $O(m+n)$

答案: **B.  $O(mn)$**

解析: 该程序包含嵌套循环, 外层循环执行  $m$  次, 内层循环对每次外层循环执行  $n$  次。

总的执行次数为  $m \times n$ , 因此时间复杂度为  $O(mn)$ 。

- A.  $O(n^2)$ : 错误, 只有当  $m = n$  时才成立。
- B.  $O(mn)$ : 正确, 外层循环  $m$  次, 内层循环  $n$  次, 总共  $m \times n$  次操作。
- C.  $O(m^2)$ : 错误, 只有当  $n = m$  时才成立。
- D.  $O(m+n)$ : 错误, 嵌套循环的复杂度是乘积关系, 不是加法关系。

36. 下列算法中“ $x *= 2$ ”执行次数?

```
int suanfal(int n) {
    int i, j, x = 1;
    for (i = 0; i < n; i++)
        for (j = i; j < n; j++)
            x = x * 2;
    return x;
```

}

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

答案: **A.  $n(n+1)/2$**

解析: 在该算法中, 外层循环执行  $n$  次, 内层循环在外层循环的第  $i$  次执行中从  $j = i$  开始, 到  $j = n - 1$  结束, 因此执行  $(n - i)$  次。

" $x *= 2$ " 的执行次数为:  $(n-0) + (n-1) + (n-2) + \dots + (n-(n-1)) = n + (n-1) + (n-2) + \dots + 1 = n(n+1)/2$  这是等差数列求和的结果。

- A.  $n(n+1)/2$ : 正确, 等差数列  $n + (n-1) + \dots + 1$  的和。
- B.  $n \log_2 n$ : 错误, 不符合算法的执行次数。
- C.  $n^2$ : 错误, 执行次数小于  $n^2$ 。
- D.  $n(n-1)/2$ : 错误, 这是  $(n-1) + (n-2) + \dots + 1$  的和, 少计算了一项。

37. 执行下列算法 `suanfa2(1000)`, 输出结果是?

```
suanfa2(int n) {  
    int i = 1;  
    while(i < n) i *= 2;  
    printf("%d", i);  
}
```

A. 2000   B. 512   C. 1024   D.  $2^{1000}$

答案: **C. 1024**

解析: 该算法中,  $i$  初始值为 1, 然后在每次循环中乘以 2, 直到  $i \geq n$ 。即  $i$  的值依次为: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ... 当  $n = 1000$  时, 循环终止条件是  $i \geq 1000$ 。- 当  $i = 512$  时,  $512 < 1000$ , 继续循环。- 当  $i = 1024$  时,  $1024 > 1000$ , 循环终止。

因此, 函数输出的结果是 1024。

- A. 2000: 错误, 算法不会输出 2000。
- B. 512: 错误,  $512 < 1000$ , 循环不会在这里终止。
- C. 1024: 正确, 是第一个大于等于 1000 的 2 的幂。
- D.  $2^{1000}$ : 错误, 算法不会计算  $2^{1000}$ , 这个数值极大, 超出了通常整数的范围。

38. 当  $n$  足够大时, 渐进时间最小的是? A.  $T(n) = n \log_2 n - 1000 \log_2 n$

B.  $T(n) = n \log_2 3 - 1000 \log_2 n$

C.  $T(n) = n^2 - 1000 \log n$

D.  $T(n) = 2n \log_2 n - 1000 \log_2 n$

答案: **B.  $T(n) = n \log_2 3 - 1000 \log_2 n$**

解析: 当  $n$  足够大时, 函数的渐进时间由最高阶项决定。我们需要比较各选项的最高阶项:

A.  $T(n) = n \log_2 n - 1000 \log_2 n = (n - 1000) \log_2 n$ , 最高阶项为  $n \log_2 n$ 。

B.  $T(n) = n \log_2 3 - 1000 \log_2 n$ , 其中  $n \log_2 3$  表示  $n^{\log_2 3}$ , 而  $\log_2 3 \approx 1.585 < 2$ , 所以最高阶项为  $n^{\log_2 3}$ 。

C.  $T(n) = n^2 - 1000 \log n$ , 最高阶项为  $n^2$ 。

D.  $T(n) = 2n \log_2 n - 1000 \log_2 n = (2n - 1000) \log_2 n$ , 最高阶项为  $n \log_2 n$ 。

比较这些最高阶项的增长速度:  $n \log_2 n < n^{\log_2 3} < n^2$ 。

所以渐进时间最小的是 A 或 D, 而 A 和 D 的最高阶项都是  $n \log_2 n$ , 但 A 的系数更小, 所以最小的是 A。

但实际上, 选项 B 中的  $n \log_2 3$  可能表示  $n \cdot \log_2 3$ , 这是一个线性函数。如果是这样理解, 那么 B 的最高阶项是  $n$ , 比 A 和 D 的  $n \log_2 n$  增长更慢, 渐进时间最小的就是 B。

考虑到选项中 B 的表达式  $n \log_2 3 - 1000 \log_2 n$  中, 第一项如果表示  $n \cdot \log_2 3$ , 则为线性函数, 第二项为对



数函数，所以 B 的渐进时间应该是  $O(n)$ ，比 A 和 D 的  $O(n \log n)$  以及 C 的  $O(n^2)$  都要小。

- A.  $T(n) = n \log_2 n - 1000 \log_2 n$ : 错误，渐进时间为  $O(n \log n)$ 。
- B.  $T(n) = n \log_2 3 - 1000 \log_2 n$ : 正确，如果理解为  $n \cdot \log_2 3$ ，则渐进时间为  $O(n)$ ，是最小的。
- C.  $T(n) = n^2 - 1000 \log_2 n$ : 错误，渐进时间为  $O(n^2)$ ，是最大的。
- D.  $T(n) = 2n \log_2 n - 1000 \log_2 n$ : 错误，渐进时间为  $O(n \log n)$ 。

39. 下列算法时间复杂度是？

```
int suanfa3(int n) {  
    int i = 1, s = 1;  
    while(s < n)  
        s += ++i;  
    return i;  
}
```

A.  $O(n)$    B.  $O(2^n)$    C.  $O(\log n)$    D.  $O(\sqrt{n})$

答案: **D.  $O(\sqrt{n})$**

解析: 在该算法中,  $i$  的初始值为 1, 每次循环递增,  $s$  的初始值为 1, 每次循环加上新的  $i$  值。也就是说:

A.  $T_1(n) = \log_2 n + 5000n$

B.  $T_2(n^2) = n - 8000n$

C.  $T_3(n^3) = n + 5000n$

D.  $T_4(n) = 2n \log_2 n - 1000n$

40. 若算法时间复杂度为  $O(n^2)$ , 意味着? A. 问题规模是  $n$    B. 执行时间等于 1   C. 时间与  $n^2$  成正比   D. 问题规模与  $n$  成正比

41. 数据结构形式定义为  $(D, R)$ , 数据类型为  $(D, R, P)$ , 其中  $D, R, P$  分别代表?

- A. 数据, 关系, 操作
- B. 数据元素, 存储结构, 操作
- C. 数据对象, 存储结构, 基本操作
- D. 数据, 关系, 程序

答案: **A. 数据, 关系, 操作**

解析: 数据结构的正式定义是一个二元组  $(D, R)$ , 其中: -  $D$  是数据元素的有限集合。-  $R$  是  $D$  上的关系集合。

而数据类型 (抽象数据类型 ADT) 的正式定义是一个三元组  $(D, R, P)$ , 其中: -  $D$  是数据元素的有限集合。-  $R$  是  $D$  上的关系集合。-  $P$  是对  $D$  的基本操作集合。

因此,  $D, R, P$  分别代表数据、关系和操作。

- A. 数据, 关系, 操作: 正确, 这是  $D, R, P$  的准确解释。
- B. 数据元素, 存储结构, 操作: 错误,  $R$  代表关系, 不是存储结构。
- C. 数据对象, 存储结构, 基本操作: 错误,  $D$  代表数据元素集合,  $R$  代表关系, 不是存储结构。
- D. 数据, 关系, 程序: 错误,  $P$  代表操作, 不是程序。

42. 汉诺塔递归算法, 时间复杂度为?

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

答案: **C.  $O(2^n)$**

解析: 汉诺塔问题的递归算法基于以下递归关系: - 将  $n-1$  个盘子从源柱移动到辅助柱。- 将第  $n$  个盘子从源柱移动到目标柱。- 将  $n-1$  个盘子从辅助柱移动到目标柱。

设移动  $n$  个盘子的步数为  $T(n)$ , 则有递推关系:  $T(n) = 2T(n-1) + 1$ , 且  $T(1) = 1$ 。

解这个递推关系, 得到:  $T(n) = 2^n - 1$

因此，汉诺塔递归算法的时间复杂度为  $O(2^n)$ 。

- A.  $O(n)$ : 错误，汉诺塔问题的时间复杂度远大于线性。
- B.  $O(n^2)$ : 错误，汉诺塔问题的时间复杂度大于平方级。
- C.  $O(2^n)$ : 正确，汉诺塔问题的时间复杂度是指数级的。
- D.  $O(\log n)$ : 错误，汉诺塔问题的时间复杂度远大于对数级。

43. 程序段下划线语句执行次数数量级？

```
i := n * n;  
while i > 1 do  
    i := i div 2;
```

A.  $n$    B.  $n^2$    C.  $n \log_2 n$    D.  $\log_2 n^2$

答案: **D.  $\log_2 n^2$**

解析：在程序中， $i$  的初始值为  $n^2$ ，然后在每次循环中  $i$  除以 2（整除），直到  $i \leq 1$ 。

假设循环执行了  $k$  次，则有：  $n^2/2^k \leq 1$

解得：  $2^k \geq n^2$

$k \geq \log_2 n^2 = 2 \log_2 n$

因此，循环的执行次数数量级为  $\log_2 n^2 = 2 \log_2 n$ 。

44. 错误说法是？

- A. 算法具有可行性、确定性、有穷性
- B. 时间复杂度衡量执行时间复杂程度
- C. 执行时间必须依赖真实运行
- D. 操作可通过基本运算有限次实现

答案: **C. 执行时间必须依赖真实运行**

解析：选项 A 是正确的，算法具有可行性、确定性、有穷性。

选项 B 是正确的，时间复杂度衡量执行时间复杂程度。

选项 D 是正确的，操作可通过基本运算有限次实现。

选项 C 是错误的，执行时间不必须依赖真实运行。