



“十二五”普通高等教育本科国家级规划教材



高等学校数据结构课程系列教材

数据结构教程 (第4版)

学习指导

李春葆 主编



清华大学出版社

高等学校数据结构课程系列教材

数据结构教程(第4版)学习指导

李春葆 主编

尹为民 蒋晶珏 喻丹丹 安 杨 编著

清华大学出版社
北京

内 容 简 介

本书是与《数据结构教程》(第4版)(李春葆等编著,清华大学出版社出版)配套的学习辅导书。两书章次一一对应,内容包括绪论、线性表、栈和队列、串、递归、数组和广义表、树形结构、图、查找、内排序、外排序和文件。各章中除给出本章练习题的参考答案外,还总结了本章的知识体系结构,并补充了大量的练习题并予以解析。附录中给出了几份近年来本科生、研究生数据结构考试试题及参考答案。书中列出了全部的练习题,因此自成一体,可以脱离主教材单独使用。

本书适合高等院校计算机及相关专业本科生及研究生使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构教程(第4版)学习指导/李春葆主编. —北京:清华大学出版社,2013.1

高等学校数据结构课程系列教材

ISBN 978-7-302-25706-6

I. ①数… II. ①李… III. ①数据结构—高等学校—习题集 IV. ①TP311.12-44

中国版本图书馆CIP数据核字(2011)第106110号

责任编辑:魏江江 薛 阳

封面设计:杨 兮

责任校对:焦丽丽

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18 字 数:440千字

版 次:2013年1月第1版 印 次:2013年1月第1次印刷

印 数:1~3000

定 价:29.00元

产品编号:041578-01

第 1 章 绪论	1
1.1 本章知识体系结构	1
1.2 教材中练习题及参考答案	2
1.3 补充练习题及参考答案	4
1.3.1 单项选择题	4
1.3.2 填空题	7
1.3.3 判断题	7
1.3.4 简答题	8
1.3.5 算法设计及算法分析题	9
第 2 章 线性表	13
2.1 本章知识体系结构	13
2.2 教材中练习题及参考答案	13
2.3 补充练习题及参考答案	21
2.3.1 单项选择题	21
2.3.2 填空题	22
2.3.3 判断题	23
2.3.4 简答题	24
2.3.5 算法设计题	25
第 3 章 栈和队列	36
3.1 本章知识体系结构	36
3.2 教材中练习题及参考答案	36
3.3 补充练习题及参考答案	41
3.3.1 单项选择题	41
3.3.2 填空题	44
3.3.3 判断题	44

3.3.4	简答题	46
3.3.5	算法设计题	48
第4章	串	60
4.1	本章知识体系结构	60
4.2	教材中练习题及参考答案	60
4.3	补充练习题及参考答案	62
4.3.1	单项选择题	62
4.3.2	填空题	63
4.3.3	判断题	63
4.3.4	简答题	63
4.3.5	算法设计题	67
第5章	递归	72
5.1	本章知识体系结构	72
5.2	教材中练习题及参考答案	72
5.3	补充练习题及参考答案	75
5.3.1	单项选择题	75
5.3.2	填空题	76
5.3.3	判断题	76
5.3.4	简答题	77
5.3.5	算法设计题	77
第6章	数组和广义表	84
6.1	本章知识体系结构	84
6.2	教材中练习题及参考答案	85
6.3	补充练习题及参考答案	87
6.3.1	单项选择题	87
6.3.2	填空题	89
6.3.3	判断题	90
6.3.4	简答题	90
6.3.5	算法设计题	92
第7章	树形结构	97
7.1	本章知识体系结构	97
7.2	教材中练习题及参考答案	98
7.3	补充练习题及参考答案	104
7.3.1	单项选择题	104
7.3.2	填空题	109

7.3.3	判断题.....	111
7.3.4	简答题.....	112
7.3.5	算法设计题.....	121
第 8 章	图.....	135
8.1	本章知识体系结构	135
8.2	教材中练习题及参考答案	136
8.3	补充练习题及参考答案	145
8.3.1	单项选择题.....	145
8.3.2	填空题.....	149
8.3.3	判断题.....	150
8.3.4	简答题.....	152
8.3.5	算法设计题.....	160
第 9 章	查找.....	171
9.1	本章知识体系结构	171
9.2	教材中练习题及参考答案	172
9.3	补充练习题及参考答案	180
9.3.1	单项选择题.....	180
9.3.2	填空题.....	182
9.3.3	判断题.....	182
9.3.4	简答题.....	184
9.3.5	算法设计题.....	188
第 10 章	内排序	195
10.1	本章知识体系结构	195
10.2	教材中练习题及参考答案	195
10.3	补充练习题及参考答案	200
10.3.1	单项选择题	200
10.3.2	填空题	203
10.3.3	判断题	204
10.3.4	简答题	205
10.3.5	算法设计题	210
第 11 章	外排序	222
11.1	本章知识体系结构	222
11.2	教材中练习题及参考答案	222
11.3	补充练习题及参考答案	224
11.3.1	单项选择题	224

11.3.2 填空题	224
11.3.3 判断题	225
11.4.4 简答题	225
第12章 文件	228
12.1 本章知识体系结构	228
12.2 教材中练习题及参考答案	228
12.3 补充练习题及参考答案	231
12.3.1 单项选择题	231
12.3.2 填空题	233
12.3.3 判断题	233
12.3.4 简答题	234
附录 A 四份本科生数据结构期末考试试题及参考答案	236
附录 B 三份数据结构考研试题及参考答案	251
附录 C 2009年全国计算机专业硕士学位研究生入学考试数据结构部分 试题及参考答案	263
附录 D 2010年全国计算机专业硕士学位研究生入学考试数据结构部分 试题及参考答案	269
附录 E 2011年全国计算机专业硕士学位研究生入学考试数据结构部分 试题及参考答案	274

绪 论

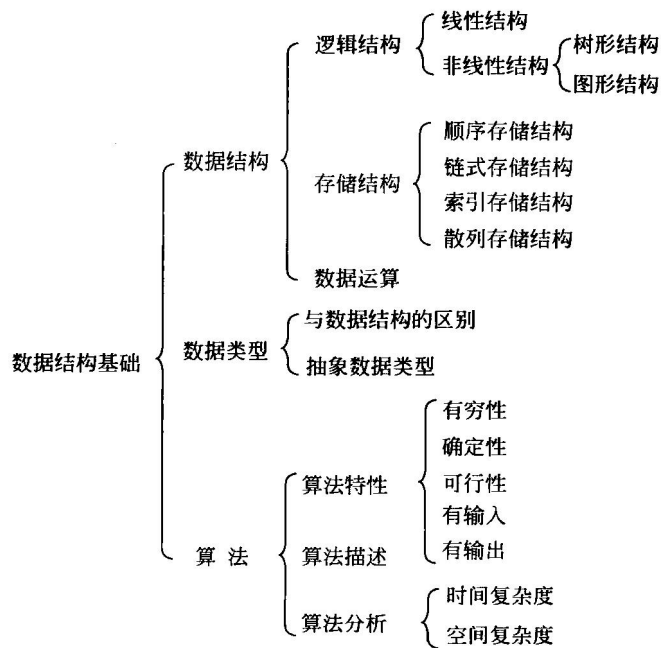
第 1 章

基本知识点：数据结构和算法的概念。

重点：数据结构的逻辑结构、存储结构、数据运算三方面的概念及相互关系；算法时间复杂度分析。

难点：分析算法的时间复杂度。

1.1 本章知识体系结构



1.2 教材中练习题及参考答案

1.1 简述数据与数据元素的关系与区别。

答:凡是能被计算机存储、加工的对象统称为数据,数据是一个集合。数据元素是数据的基本单位,是数据的个体。数据元素与数据之间的关系是元素与集合之间的关系。

1.2 数据结构和数据类型有什么区别?

答: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,一般包括三个方面的内容,即数据的逻辑结构、存储结构和数据的运算。而数据类型是一个值的集合和定义在这个集合上的一组运算的总称,如C语言中的int数据类型是由-32768~32767(16位机)的整数和+、-、*、/、%等运算符构成。

1.3 设3个表示算法频度的函数 $f(n)$ 、 $g(n)$ 和 $h(n)$ 分别为:

$$f(n) = 100n^3 + n^2 + 1000$$

$$g(n) = 25n^3 + 5000n^2$$

$$h(n) = n^{1.5} + 5000n \log_2 n$$

求它们对应的时间复杂度。

答: $f(n) = 100n^3 + n^2 + 1000 = O(n^3)$, $g(n) = 25n^3 + 5000n^2 = O(n^3)$

当 $n \rightarrow \infty$ 时, $\sqrt{n} > \log_2 n$, 所以 $h(n) = n^{1.5} + 5000n \log_2 n = O(n^{1.5})$ 。

1.4 用C/C++语言描述下列算法,并给出算法的时间复杂度。

(1) 求一个n阶方阵的所有元素之和。

(2) 对于输入的任意三个整数,将它们按从小到大的顺序输出。

(3) 对于输入的任意n个整数,输出其中的最大和最小元素。

答:(1) 算法如下:

```
int sum(int A[n][n], int n)
{   int i, j, s = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            s = s + A[i][j];
    return(s);
}
```

本算法的时间复杂度为 $O(n^2)$ 。

(2) 算法如下:

```
void order(int a, int b, int c)
{   if (a > b)
    {   if (b > c)
        printf("%d, %d, %d\n", c, b, a);
        else if (a > c)
            printf("%d, %d, %d\n", b, c, a);
        else
            printf("%d, %d, %d\n", b, a, c);
    }
    else
```

```

    {   if (b>c)
        {   if (a>c)
                printf(" %d, %d, %d\n",c,a,b);
            else
                printf(" %d, %d, %d\n",a,c,b);
        }
        else printf(" %d, %d, %d\n",a,b,c);
    }
}

```

本算法的时间复杂度为 $O(1)$ 。

(3) 算法如下:

```

void maxmin(int A[], int n, int &max, int &min)
{   int i;
    min = max = A[0];
    for (i = 1; i < n; i++)
    {   if (A[i] > max)   max = A[i];
        if (A[i] < min)   min = A[i];
    }
}

```

本算法的时间复杂度为 $O(n)$ 。

1.5 设 n 为正整数, 给出下列各种算法关于 n 的时间复杂度。

(1)

```

void fun1(int n)
{   i = 1, k = 100;
    while (i < n)
    {   k = k + 1;
        i += 2;
    }
}

```

(2)

```

void fun2(int b[], int n)
{   int i, j, k, x;
    for (i = 0; i < n - 1; i++)
    {   k = i;
        for (j = i + 1; j < n; j++)
            if (b[k] > b[j]) k = j;
        x = b[i]; b[i] = b[k]; b[k] = x;
    }
}

```

(3)

```

void fun3(int n)
{   int i = 0, s = 0;
    while (s < n)
    {   i++;

```

```

        s = s + i;
    }
}

```

答: (1) 设 while 循环语句执行次数为 $T(n)$, 则:

$$i = 2T(n) + 1 \leq n - 1, \text{ 即 } T(n) \leq \frac{n}{2} - 1 = O(n).$$

(2) 算法中的基本运算语句是 $\text{if}(b[k] > b[j]) k=j$, 其执行次数 $T(n)$ 为:

$$T(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-i-1) = \frac{n(n-1)}{2} = O(n^2)$$

(3) 设 while 循环语句执行次数为 $T(n)$, 则:

$$s = 1 + 2 + \dots + T(n) = \frac{T(n)(T(n)+1)}{2} \leq n-1, \text{ 则 } T(n) = O(\sqrt{n}).$$

1.6 有以下递归算法用于对数组 $a[i..j]$ 的元素进行归并排序:

```

void mergesort(int a[], int i, int j)
{
    int m;
    if (i != j)
    {
        m = (i + j) / 2;
        mergesort(a, i, m);
        mergesort(a, m + 1, j);
        merge(a, i, j, m);
    }
}

```

求 $\text{mergesort}(a, 0, n-1)$ 的时间复杂度。其中, $\text{merge}(a, i, j, m)$ 用于两个有序子序列 $a[i..m]$ 和 $a[m+1..j]$ 的合并, 是非递归函数, 它的时间复杂度为 $O(\text{合并的元素个数})$ 。

答: 设 $\text{mergesort}(a, 0, n-1)$ 的执行次数为 $T(n)$, 分析得到以下递归关系:

$$T(n) = \begin{cases} O(1) & n = 1 \\ 2T\left(\frac{n}{2}\right) + O(n) & n > 1 \end{cases}$$

$O(n)$ 为 $\text{merge}()$ 所需的时间, 设为 cn (c 为常量)。因此:

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + cn = 2\left(2T\left(\frac{n}{2^2}\right) + \left(\frac{cn}{2}\right)\right) + cn = 2^2 T\left(\frac{n}{2^2}\right) + 2cn = 2^3 T\left(\frac{n}{2^3}\right) + 3cn \\ &\vdots \\ &= 2^k T\left(\frac{n}{2^k}\right) + kcn = 2^k O(1) + kcn \end{aligned}$$

由于 $\frac{n}{2^k}$ 趋近于 1, 则 $k = \log_2 n$ 。

所以 $T(n) = 2^{\log_2 n} O(1) + cn \log_2 n = n + cn \log_2 n = O(n \log_2 n)$ 。

1.3 补充练习题及参考答案

1.3.1 单项选择题

1. 数据结构是一门研究程序设计中数据的 ① 以及它们之间的 ② 和运算等的学科。

- ① A. 元素 B. 计算方法 C. 逻辑存储 D. 映像
 ② A. 结构 B. 关系 C. 运算 D. 算法

答: ① A ② B。

2. 在数据结构中,从逻辑上可以把数据结构分为_____两类。
 A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
 C. 线性结构和非线性结构 D. 内部结构和外部结构

答: C。

3. 数据的逻辑结构是_____关系的整体。
 A. 数据元素之间逻辑 B. 数据项之间逻辑
 C. 数据类型之间 D. 存储结构之间

答: A。

4. 在计算机的存储器中表示数据时,物理地址和逻辑地址的相对位置相同并且是连续的,称之为_____。

- A. 逻辑结构 B. 顺序存储结构
 C. 链式存储结构 D. 以上都对

答: B。

5. 在链式存储结构中,通常一个存储节点用于存储一个_____。
 A. 数据项 B. 数据元素 C. 数据结构 D. 数据类型

答: B。

6. 数据运算的执行_____。
 A. 效率与采用何种存储结构有关 B. 是根据存储结构来定义的
 C. 有算术运算和关系运算两大类 D. 必须用程序设计语言来描述

答: A。

7. 数据结构在计算机内存中的表示是指_____。
 A. 数据的存储结构 B. 数据结构
 C. 数据的逻辑结构 D. 数据元素之间的关系

答: A。

8. 在数据结构中,与所使用的计算机无关的是_____。
 A. 逻辑结构 B. 存储结构
 C. 逻辑结构和存储结构 D. 物理结构

答: A。

9. 数据采用链式存储结构存储,要求_____。
 A. 每个节点占用一片连续的存储区域
 B. 所有节点占用一片连续的存储区域
 C. 节点的最后一个数据域是指针类型
 D. 每个节点有多少个后继,就设多少个指针域

答: A。

10. 下列说法中,不正确的是_____。
 A. 数据元素是数据的基本单位

- B. 数据项是数据中不可分割的最小可标识单位
- C. 数据可由若干个数据元素构成
- D. 数据项可由若干个数据元素构成

答: 数据元素可由若干个数据项构成。本题答案为 D。

11. 以下_____不是算法的基本特性。

- A. 可行性
- B. 长度有限
- C. 在确定的时间内完成
- D. 确定性

答: 选项 C 指的是有穷性, 属算法的基本特性。本题答案为 B。

12. 在计算机中算法指的是解决某一问题的有限运算序列, 它必须具备输入、输出、_____。

- A. 可行性、可移植性和可扩充性
- B. 可行性、有穷性和确定性
- C. 确定性、有穷性和稳定性
- D. 易读性、稳定性和确定性

答: B。

13. 下面关于算法的说法正确的是_____。

- A. 算法最终必须由计算机程序实现
- B. 一个算法所花时间等于该算法中每条语句的执行时间之和
- C. 算法的可行性是指指令不能有二义性
- D. 以上说法都是错误的

答: 算法最终不一定由计算机程序实现, 算法的确定性是指不能有二义性。本题答案为 B。

14. 算法的时间复杂度与_____有关。

- A. 问题规模
- B. 计算机硬件性能
- C. 编译程序质量
- D. 程序设计语言

答: A。

15. 算法分析的主要任务之一是分析_____。

- A. 算法是否具有较好的可读性
- B. 算法中是否存在语法错误
- C. 算法的功能是否符合设计要求
- D. 算法的执行时间和问题规模之间的关系

答: D。

16. 算法分析的目的是_____。

- A. 找出数据结构的合理性
- B. 研究算法中输入和输出关系
- C. 分析算法的效率以求改进
- D. 分析算法的易读性和文档性

答: 算法分析即算法效率分析, 包括时间复杂度和空间复杂度分析, 其目的是为了改进算法效率。本题答案为 C。

17. 某算法的时间复杂度为 $O(n^2)$, 表明该算法的_____。

- A. 问题规模是 n^2
- B. 执行时间等于 n^2
- C. 执行时间与 n^2 成正比
- D. 问题规模与 n^2 成正比

答: C。

1.3.2 填空题

1. 数据的逻辑结构是指_____。

答：数据元素之间的逻辑关系。

2. 一个数据结构在计算机中的_____称为存储结构。

答：映像。

3. 顺序存储方法是把逻辑上 ① 存储在物理位置上 ② 里；链式存储方法中节点间的逻辑关系是由 ③ 的。

答：① 相邻的节点 ② 相邻的存储单元 ③ 附加的指针字段表示。

4. 一个算法具有 5 个特性：_____、_____、_____、输入和输出。

答：可行性、有穷性、确定性

5. 算法的执行时间是_____的函数。

答：问题规模。

6. 以下为各算法所有语句频度之和的表达式，其中时间复杂度相同的是_____。

A. $T_A(n) = 2n^3 + 3n^2 + 1000$

B. $T_B(n) = n^3 - n^2 \log_2 n - 1000$

C. $T_C(n) = n^2 \log_2 n + n^2$

D. $T_D(n) = n^2 + 1000$

答： $T_A(n) = O(n^3)$, $T_B(n) = O(n^3)$, $T_C(n) = O(n^2 \log_2 n)$, $T_D(n) = O(n^2)$, 所以，时间复杂度相同的是 A 和 B。

1.3.3 判断题

1. 判断以下叙述的正确性。

(1) 数据元素是数据的最小单位。

(2) 数据对象就是一组数据元素的集合。

(3) 任何数据结构都具备三个基本运算：插入、删除和查找。

(4) 数据对象是由有限个类型相同的数据元素构成的。

(5) 数据的逻辑结构与各数据元素在计算机中如何存储有关。

(6) 如果数据元素值发生改变，则数据的逻辑结构也随之改变。

(7) 逻辑结构相同的数据，可以采用多种不同的存储方法。

(8) 逻辑结构不相同的数据，必须采用不同的存储方法来存储。

(9) 数据的逻辑结构是指数据元素的各数据项之间的逻辑关系。

答：

(1) 数据项是数据的最小单位。错误。

(2) 这里未强调数据元素的性质相同。错误。

(3) 如队列和栈等数据结构并不具备查找运算。错误。

(4) 正确。

(5) 错误。

(6) 错误。

(7) 正确。

(8) 错误。

(9) 数据的逻辑结构是指数据的各数据元素之间的逻辑关系。错误。

2. 判断以下叙述的正确性。

(1) 顺序存储方式只能用于存储线性结构。

(2) 数据元素是数据的最小单位。

(3) 算法可以用不同的语言描述,如果用C语言或Pascal语言等高级语言来描述,则算法就等同于程序。

(4) 数据结构是带有结构的数据元素的集合。

(5) 数据的逻辑结构是指各数据元素之间的逻辑关系。

(6) 数据结构、数据元素、数据项在计算机中的映像(或表示)分别称为存储结构、节点和数据域。

(7) 数据的物理结构是指数据在计算机内的实际的存储形式。

答:

(1) 错误。顺序存储方式也可用来存储树形结构,如完全二叉树可用一维数组存储。

(2) 错误。数据元素是数据的基本单位,数据元素可以由数据项组成。

(3) 错误。算法用各种计算机语言描述则表现为一个程序逻辑但并不等于程序,因为程序逻辑不一定满足有穷性。

(4) 正确。

(5) 正确。

(6) 正确。

(7) 正确。

1.3.4 简答题

1. 数据运算是数据结构的一个重要方面。试举一例说明两个数据结构的逻辑结构和存储方式完全相同,只是对于运算的定义不同,因而两个结构具有显著不同的特性,是两个不同的数据结构。

答:在数据结构中这类例子较多,如线性表和栈等。下面以二叉树和二叉排序树进行说明。

二叉树的定义为:二叉树是由有限的节点构成的,或者是空,或者由一个根节点和两棵互不相交的称为左子树和右子树的二叉树组成。

二叉排序树的定义为:二叉排序树或者是空树,或者是满足如下性质的二叉树:

(1) 若它的左子树非空,则左子树上所有记录的值均小于根记录的值。

(2) 若它的右子树非空,则右子树上所有记录的值均大于根记录的值。

(3) 左、右子树本身又各是一棵二叉排序树。

两者在逻辑结构和存储方式完全相同,因为二叉排序树完全可以采用二叉树的逻辑表示和存储方式。两者的运算有建立树和查找节点等。对于二叉树和二叉排序树,这些运算的定义是不同的,以查找节点为例,二叉树的时间复杂度为 $O(n)$,而二叉排序树的时间复杂度为 $O(h)$, h 为二叉排序树的高度。

二叉树和二叉排序树是两个不同的数据结构。前者通常用于表示层次关系,后者通常用于排序和查找。

2. 当为解决某一问题而选择数据的存储结构时,应从哪些方面考虑?

答:通常从两方面考虑:算法所需的存储空间量和算法所需的执行时间。

3. 按增长率由小至大的顺序排列下列各函数:

$$2^{100}, (2/3)^n, (3/2)^n, n^n, n!, 2^n, \log_2 n, n^{\log_2 n}, n^{3/2}, \sqrt{n}$$

答:按增长率由小至大的顺序可排列如下:

$$(2/3)^n < 2^{100} < \log_2 n < \sqrt{n} < n^{3/2} < n^{\log_2 n} < (3/2)^n < 2^n < n! < n^n$$

1.3.5 算法设计及算法分析题

1. 分析以下算法的时间复杂度。

```
void fun(int n)
{   int y = 0;
    while (y * y <= n)
        y++;
}
```

答:设 while 语句的执行频度为 $T(n)$,每循环一次 y 增大 1,即 $y = T(n)$,有:

$$T(n) \times T(n) \leq n \text{ 即 } T(n)^2 \leq n$$

$$T(n) \leq \sqrt{n} = O(\sqrt{n})$$

2. 分析以下算法的时间复杂度。

```
void fun(int n)
{   int i, x = 0;
    for (i = 1; i < n; i++)
        for (j = i + 1; j <= n; j++)
            x++;
}
```

答:设 $x++$ 语句执行次数为 $T(n)$,则:

$$T(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n 1 = \sum_{i=1}^{n-1} (n-i) = n(n-1)/2 = O(n^2)$$

该算法的时间复杂度为 $O(n^2)$ 。

3. 分析以下算法的时间复杂度。

```
void fun(int n)
{   int s = 0, i, j, k;
    for (i = 0; i <= n; i++)
        for (j = 0; j <= i; j++)
            for (k = 0; k < j; k++)
                s++;
}
```

答:该算法的基本运算是语句 $s++$,其频度:

$$T(n) = \sum_{i=0}^n \sum_{j=0}^i \sum_{k=0}^{j-1} 1 = \sum_{i=0}^n \sum_{j=0}^i (j-1-0+1) = \sum_{i=0}^n \sum_{j=0}^i j$$

$$\begin{aligned}
 &= \sum_{i=0}^n \frac{i(i+1)}{2} = \frac{1}{2} \left(\sum_{i=0}^n i^2 + \sum_{i=0}^n i \right) = \frac{1}{2} \left(\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right) \\
 &= \frac{2n^3 + 6n^2 + 4n}{12} = O(n^3)
 \end{aligned}$$

则该算法的时间复杂度为 $O(n^3)$ 。

4. 设 n 是偶数, 试计算运行下列算法后 m 的值并给出其时间复杂度。

```

void fun(int n)
{   int m=0, i, j;
    for (i=1; i<=n; i++)
        for (j=2 * i; j<=n; j++)
            m++;
}

```

答: 算法的基本运算为 $m++$, 由于内循环从 $2i$ 至 n , 即 i 的最大值满足: $2i \leq n, i \leq n/2$, 所以该语句的频度:

$$T(n) = \sum_{i=1}^{n/2} \sum_{j=2i}^n 1 = \sum_{i=1}^{n/2} (n - 2i + 1) = n * \frac{n}{2} - 2 \sum_{i=1}^{n/2} i + \frac{n}{2} = \frac{n^2}{4} = O(n^2)$$

该算法的时间复杂度为 $O(n^2)$ 。

5. 设 n 为正整数, 分析以下算法中各语句的频度。

```

void fun(int n)
{   int i, j, k;
    for (i=0; i<n; i++) //语句①
        for (j=0; j<n; j++) //语句②
            {   c[i][j]=0; //语句③
                for (k=0; k<n; k++) //语句④
                    c[i][j]=c[i][j]+a[i][k]*b[k][j]; //语句⑤
            }
}

```

解: 语句①的频度为 $n+1$ (i 从 0 到 $n-1$, 共计 n 次, 当 $i=n$ 时还要执行一次 $i<n$ 的比较, 故总共 $n+1$ 次); 语句②的频度为 $\sum_{i=0}^{n-1} (n+1) = n(n+1)$; 语句③的频度为 $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 1 = n^2$; 语句④的频度为 $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (n+1) = n^3 + n^2$; 语句⑤的频度为 $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} 1 = n^3$ 。

6. 设 n 为 3 的倍数, 分析以下算法的时间复杂度。

```

void fun(int n)
{   int i, j, x, y;
    for (i=1; i<=n; i++)
        if (3 * i <= n)
            for (j=3 * i; j<=n; j++)
                {   x++;
                    y=3 * x + 2;
                }
}

```