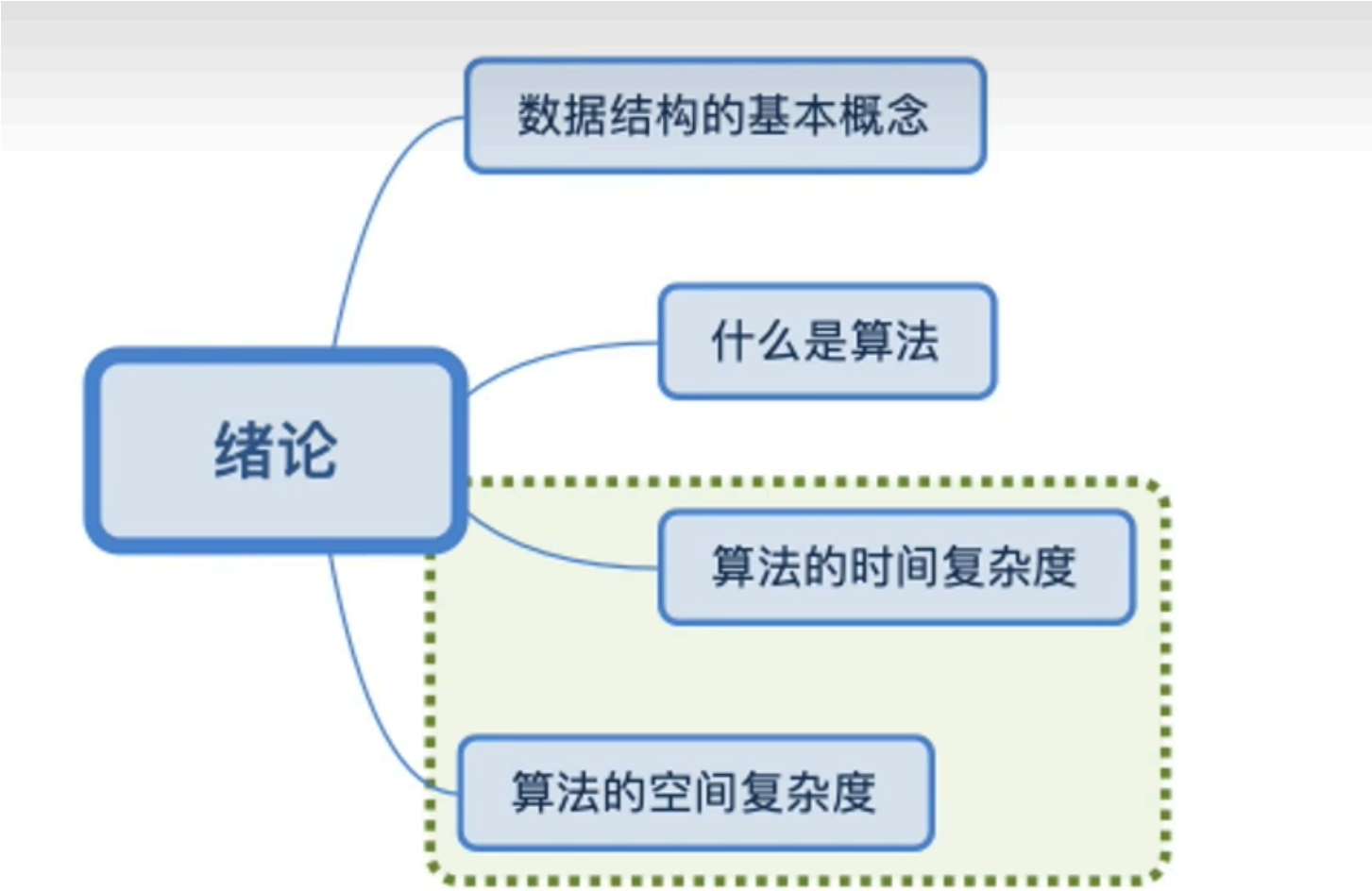
一.概论



1.1基本术语

数据结构示例：

编号	姓名	基本工资	奖金	…	…

(a) 工资表示例

序号	学号	姓名	成绩	备注

(b) 成绩表示例

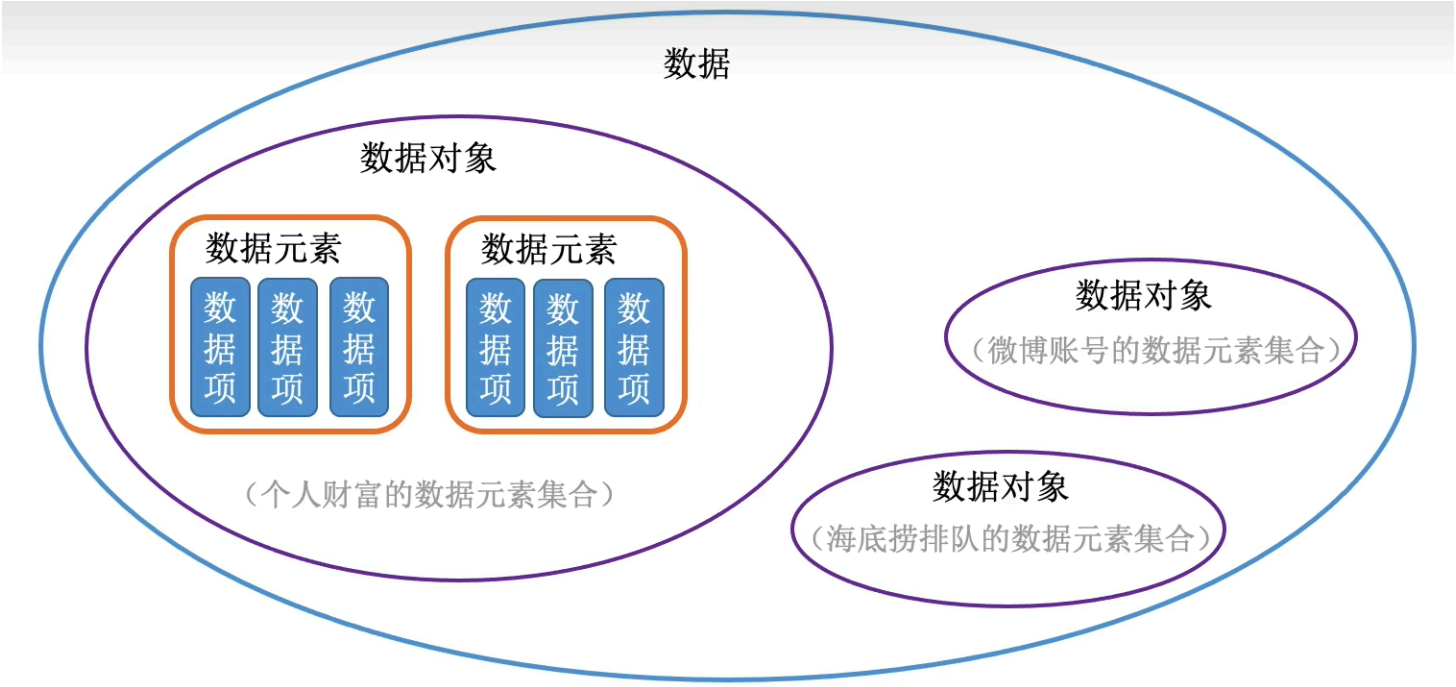
数据元素是数据的基本单位，通常作为一个整体进行考虑和处理。

一个数据元素可由若干**数据项**构成，数据项是构成数据元素的不可分割的最小单位。

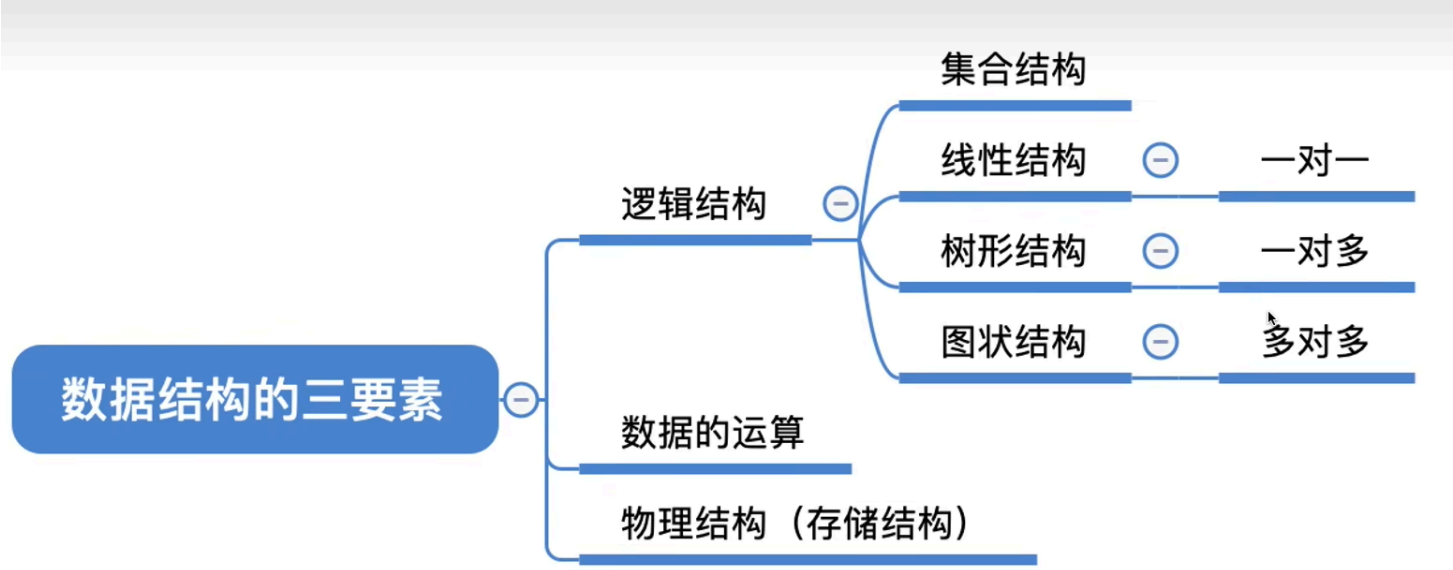
数据对象是具有相同性质的数据元素的集合。

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。

注：同一个数据对象里的数据元素，可以组成不同的数据结构。



1.2 数据结构的三要素



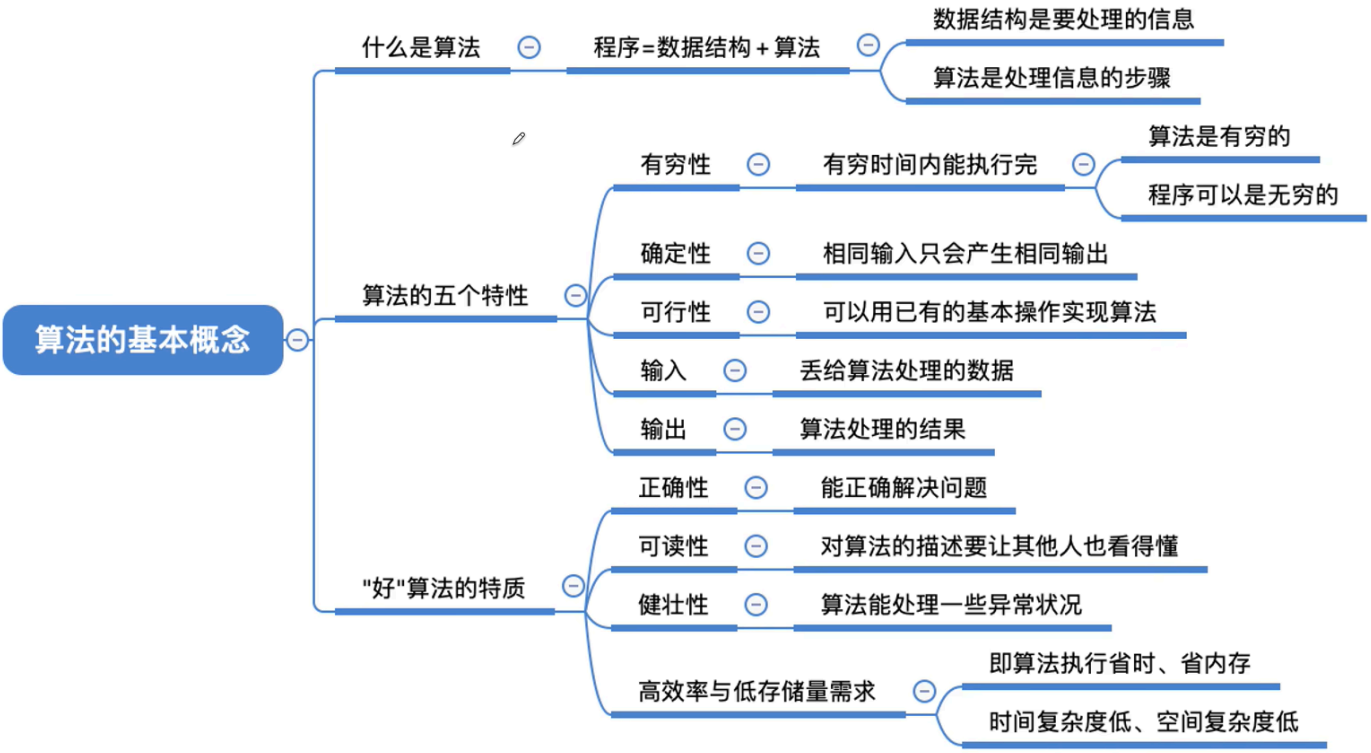
1. **逻辑结构**，定义一种数据结构；
2. **物理结构（存储结构）**，如何用计算机实现这种数据结构，表现数据元素的逻辑关系。
   - 顺序存储：把逻辑上相邻的元素存储在物理位置也相邻的存储单元中，元素之间的关系由存储单元的邻接关系来体现。
   - 链式存储：逻辑上相邻的元素在物理位置上可以不相邻，借助指示元素存储地址的指针来表示元素之间的逻辑关系。
   - 索引存储：在存储元素信息的同时，还建立附加的索引表，索引表中的每项称为索引项，索引项的一般形式是（关键字，地址）。
   - 散列存储：根据元素的关键字直接计算出该元素的存储地址，又称哈希（Hash）存储。

注：

- 若采用顺序存储，则各个数据元素在物理上必须是连续的;若采用非顺序存储，则各个数据元素在物理上可以是离散的。
- 数据的存储结构会影响存储空间分配的方便程度。
- 数据的存储结构会影响对数据运算的速度

3. **数据的运算**
   - 数据上的运算包括运算的定义和实现。
   - 运算的定义是针对逻辑结构指出运算的功能。
   - 运算的实现是针对存储结构的，指出运算的具体操作步骤。

1.3 算法的基本概念



程序 = 数据结构+算法

- 数据结构：如何用数据正确地描述现实世界的问题，并存入计算机。
- 算法：是对特定问题求解步骤的一种描述，它是指令的有限序列，其中的每条指令表示一个或多个操作。（如何高效地处理这些数据，以解决实际问题）
- 算法的特性：
  - 有穷性：一个算法必须总在执行有穷步之后结束，且每一步都可在有穷时间内完成。
  - 确定性：算法中每条指令必须有确定的含义，对于相同的输入只能得到相同的输出。
  - 可行性：算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。
  - 输入：一个算法有零个或多个输入，这些输入取自于某个特定的对象的集合。
    -输出：一个算法有一个多个输出，这些输出是与输入有着某种特定关系的量。
- 好的算法达到的目标：
  - 正确性：算法应能够正确的求解问题。
  - 可读性：算法应具有良好的可读性，以帮助人们理解。
  - 健壮性：输入非法数据时，算法能适当地做出反应或进行处理，而不会产生莫名奇妙地输出结果。
  - 效率与低存储量需求：花的时间少即：时间复杂度低。不费内存即：空间复杂度低。

1.4 时间复杂度

- 顺序执行的代码只会影响常数项，可以忽略。
- 只需挑循环中的一个基本操作分析它的执行次数与 n 的关系即可。
- 如果有多层嵌套循环只需关注最深层循环循环了几次。

```c
//算法4—— 搜索数字型爱你
void loveYou(int flag[], int n) {   //n 为问题规模
    printf("I Am Iron Man\n");
    for(int i=0; i<n; i++){ //从第一个元素开始查找
        if(flag[i]==n){ //找到元素n
            printf("I Love You %d\n", n);
            break;  //找到后立即跳出循环
        }
    }
}
```

```c
//flag 数组中乱序存放了 1~n 这些数
int flag[n]={1...n};
loveYou(flag, n);
```

很多算法执行时间与输入的数据有关

计算上述算法的时间复杂度 T(n)

最好情况：元素n在第一个位置 ——最好时间复杂度 T(n)=O(1)

最坏情况：元素n在最后一个位置 ——最坏时间复杂度 T(n)=O(n)

平均情况：假设元素n在任意一个位置的概率相同为 $\frac{1}{n}$ ——平均时间复杂度 T(n)=O(n)

循环次数 x = (1+2+3+...+n) $\frac{1}{n}$ = $(\frac{n(1+n)}{2})\frac{1}{n}$ = $\frac{1+n}{2}$　　T(n)=O(x)=O(n)

1.5 空间复杂度

- 指算法消耗的存储空间(即算法除本身所需存储外的辅助空间)
- 算法的空间复杂度S(n)定义为该算法所耗费的存储空间，它是问题规模n的函数。
  记为S(n)=O(g(n))