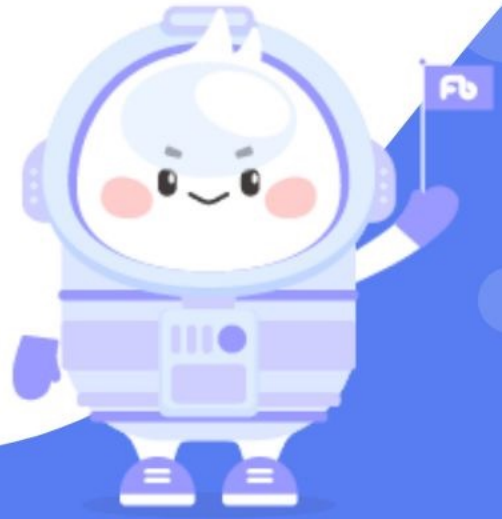
粉笔

2023上粉笔教资

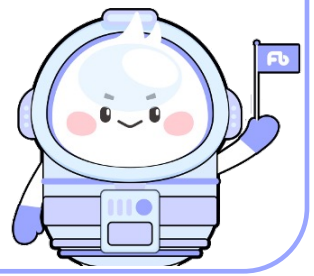
《信息技术》

数据结构与算法 2/5

讲师：孙珍珍



❀ 复习一下



四、算法的分析

（一）正确性分析

（1）一组数据

（2）苛刻数据

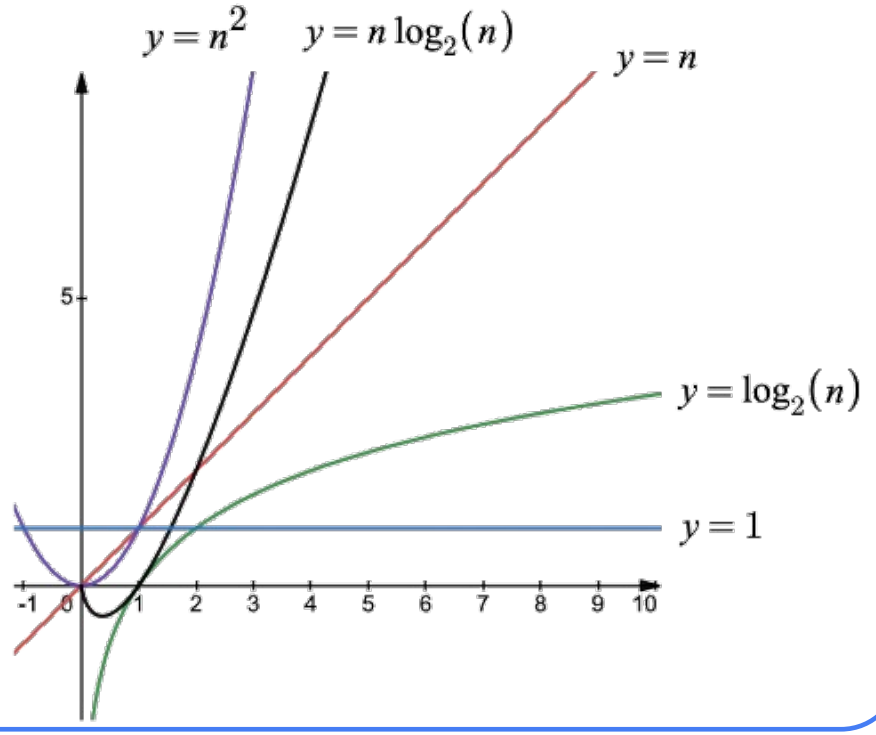
（3）一切数据

（三）空间复杂度

- 不是代码文件所占的存储空间
- 是执行时所占用的附加空间数量

（二）时间复杂度

- 不是算法执行所消耗的真正时间，是与问题规模（通过统计某语句的执行次数得出）有关
- 是一个预估值，在执行次数中去掉常数项、去掉系数、保留最高阶，用大O表示法
- 量级大小：$O(1) < O(\log_2 n) < O(n) < O(n\log_2 n) < O(n^2) < O(2^n) < O(n!) < O(n^n)$



1.某算法的时间复杂度为O(n)，表明该算法的（　　）。

A.问题规模等于n

B.执行时间等于n

C.执行时间与n成正比

D.问题规模与n成正比

2.对于同样的问题规模n，下列哪种算法时间复杂度最大（　　）。

A.O(n)

B.O(logn)

C.O(nlogn)

D.$O(n^2)$

（二）时间复杂度

1.常数阶 $O(1)$

```
int  sum = 0, n = 100 ;
sum  =  (1+n)*n/2 ;
```

2.线性阶 $O(n)$

```
int  i = 0, n = 100, sum = 0;
for ( i = 0; i < n; i++)
{
      sum = sum + i;
}
```

3.平方阶 $O(n^2)$

```
int i, j, n = 100;
for( i=0; i < n; i++ )
{
    for( j=0; j < n; j++ )
    {
        printf("hello");
    }
}
```

4.对数阶 $O(\log_2 n)$

```
int i = 1, n = 100;
while( i < n )
{
    i = i * 2;
}
```

以下算法中加下划线的语句的执行次数为（　　）。

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

A.n(n+1)　　B.n　　C.n+1　　D.n^2

第二节 数据结构基础

一、相关术语

（一）数据

- 描述客观事物的符号，即计算机可以操作的对象

（二）数据元素和数据项

- 数据元素：基本单位，一个整体，一个记录
- 数据项：数据元素的组成部分

（三）数据对象

- 相同性质的元素的集合

（四）数据结构

- 相互之间存在关系的元素集合

学号	姓名	性别	系别	年龄
11001	冯明	男	计算机	18
11002	陈月	女	通信	19
12001	褚共	男	计算机	18
12002	卫潮	男	自动化	20
12003	蒋生	女	通信	20

二、数据结构的三要素

P283

粉笔

（一）数据的逻辑结构【4】

- 元素间固有关系，与在计算机中存储位置无关

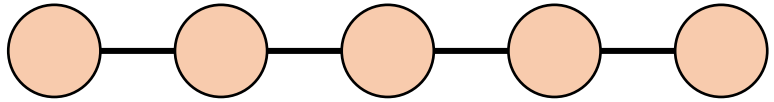
（二）数据的物理结构【2】

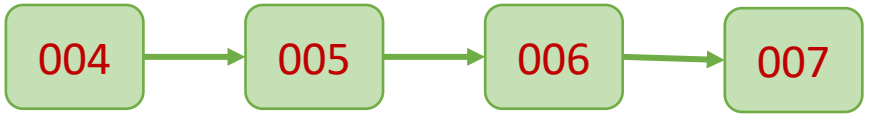
- 数据在计算机中存储方式

（三）数据的运算

- 运算的定义和实现

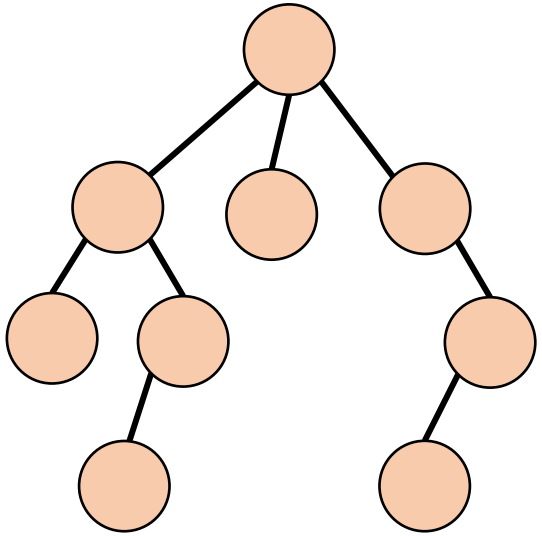
1.线性结构

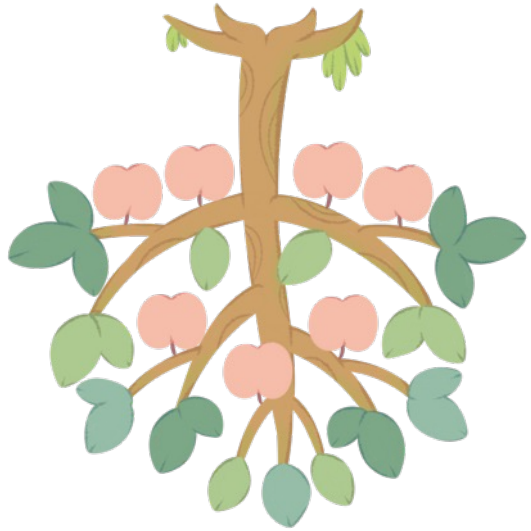


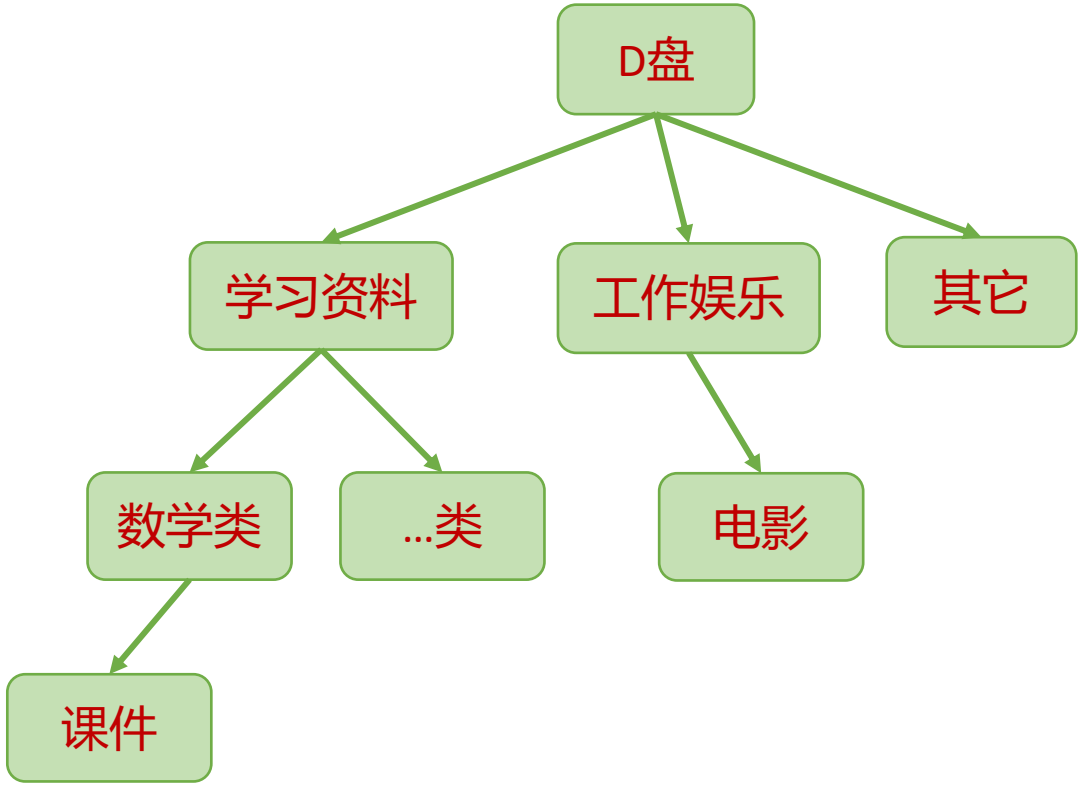


- 一对一的关系
- 只有一个开始节点和终端节点
- 每个节点最多一个前驱和后继

2.树形结构

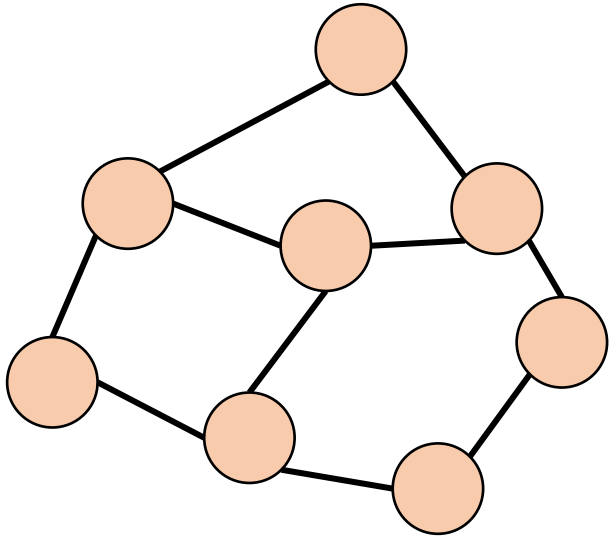


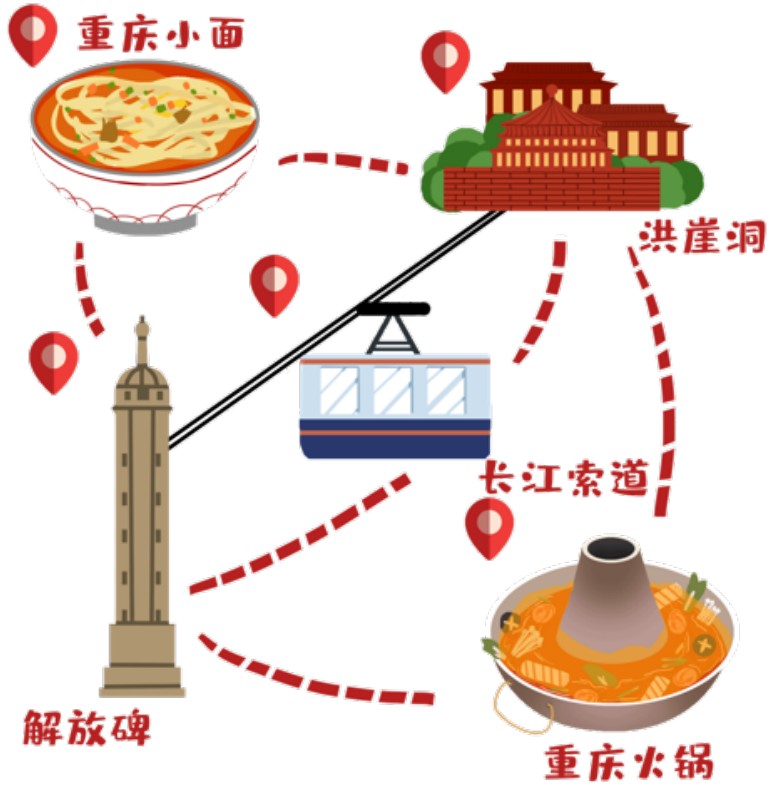




一对多的关系

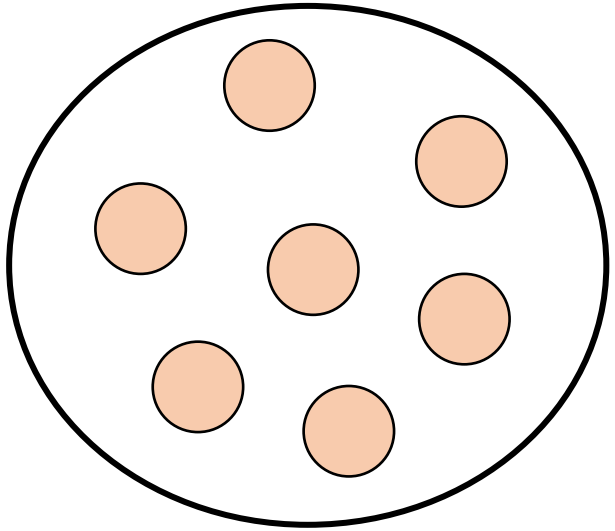
3.图形结构





多对多的关系

4.集合



无关系

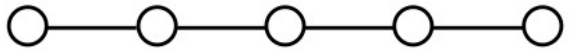
（一）数据逻辑结构

总结下

粉笔

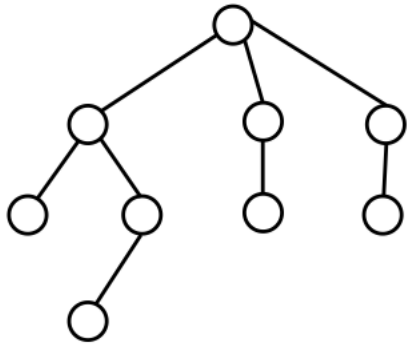
◆元素间固有关系，与在计算机中存储位置无关

1.线性结构



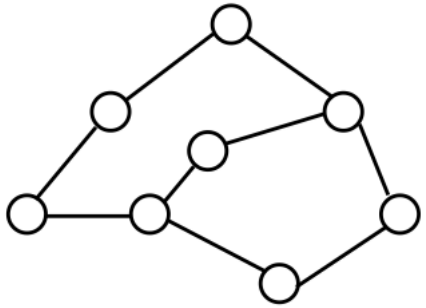
一对一

2.树形结构



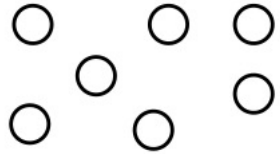
一对多

3.图形结构



多对多

4.集合



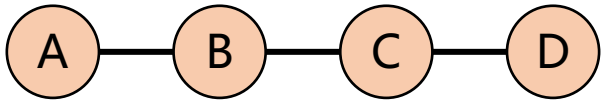
无关系

（二）数据物理结构

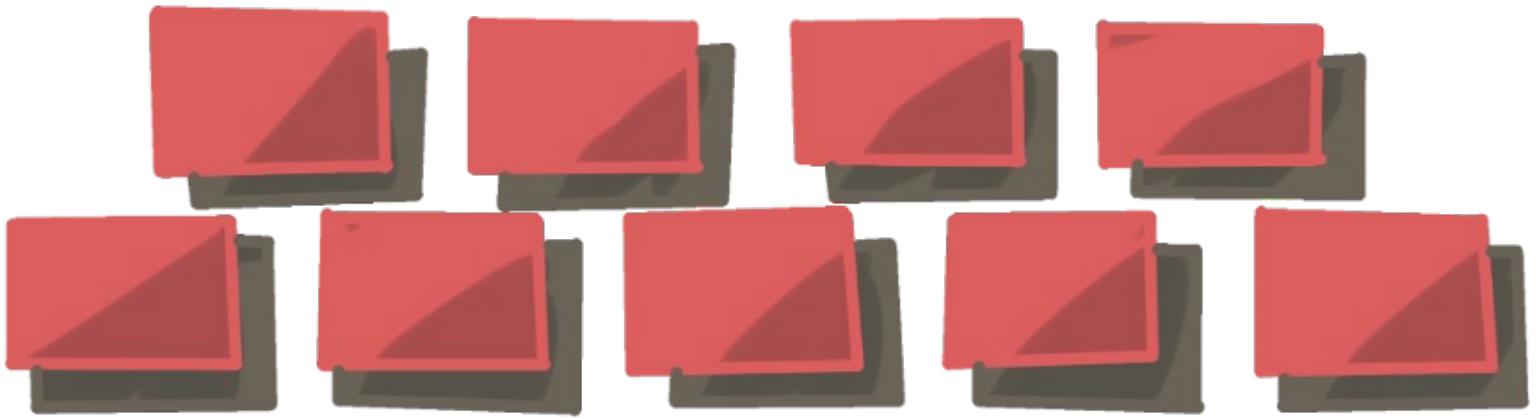
P284

粉笔

1.顺序存储结构

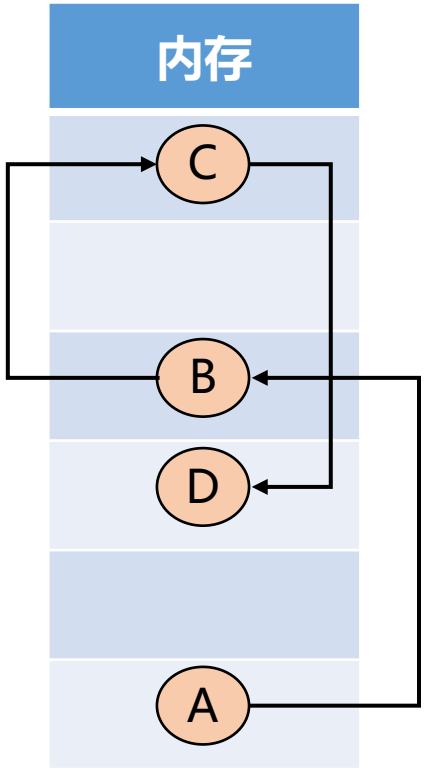


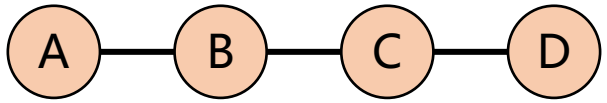
内存
A
B
C
D

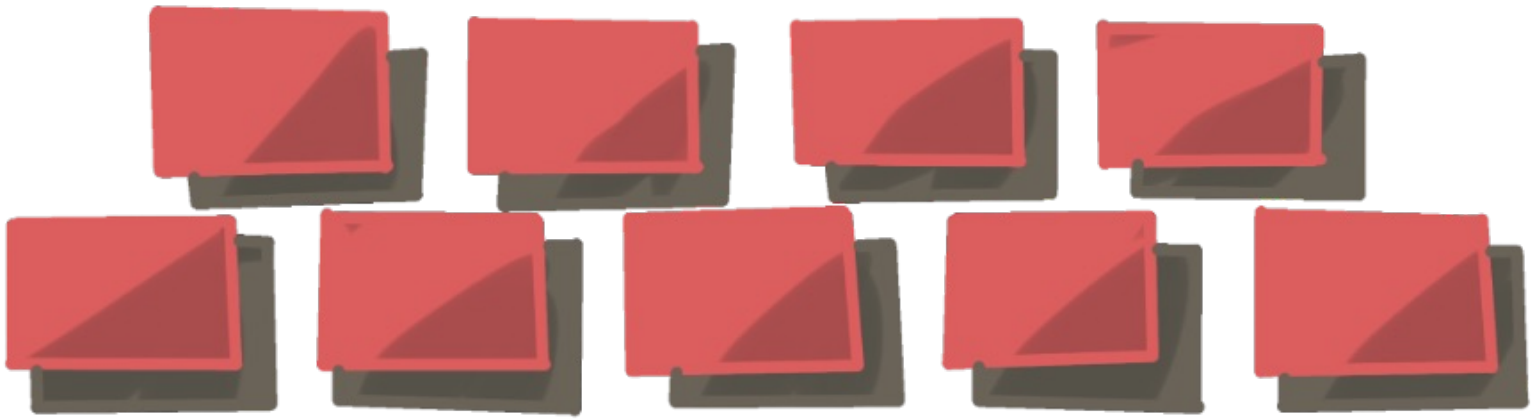


- 数据存放在地址连续的空间
- 借助相邻位置找下一个数据

2.链式存储结构







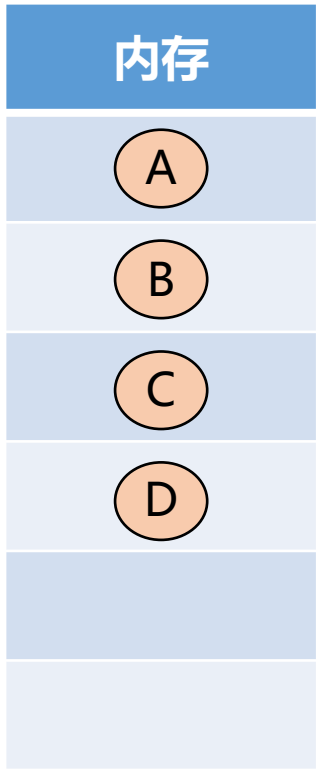
- 数据存放在任意的空间
- 借助指针查找下一个数据

（二）数据物理结构

总结下

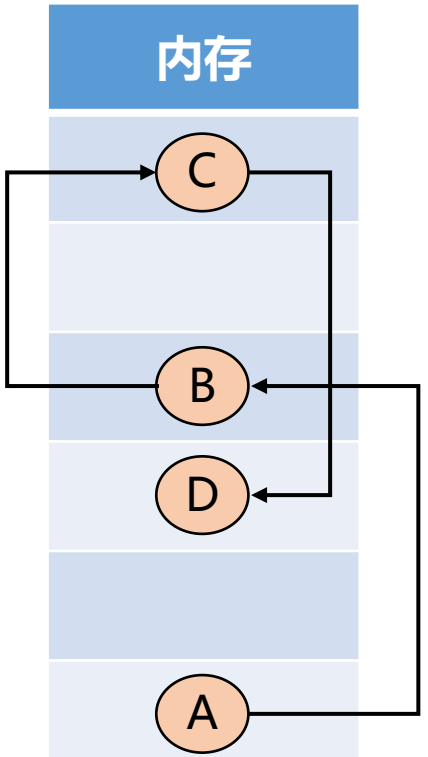
粉笔

◆数据在计算机中的存储方式



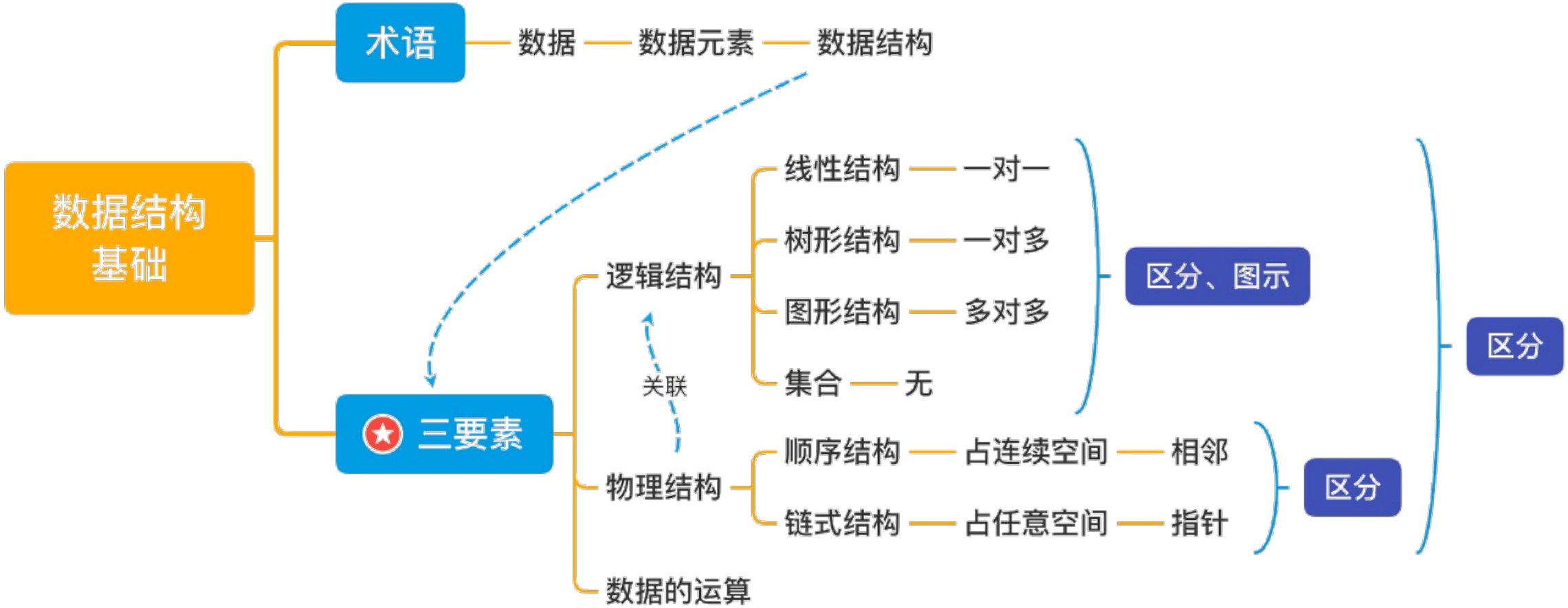
1.顺序存储结构：

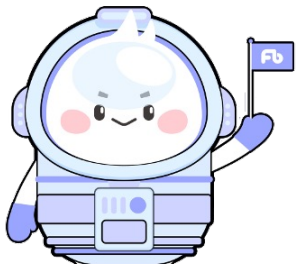
- 数据存放在地址连续的空间
- 借助相邻位置找下一个数据



2.链式存储结构：

- 数据存放在任意的空间
- 借助指针查找下一个数据


数据结构基础
术语
数据
数据元素
数据结构
三要素
逻辑结构
线性结构
一对一
树形结构
一对多
图形结构
多对多
集合
无
区分、图示
关联
物理结构
顺序结构
占连续空间
相邻
链式结构
占任意空间
指针
区分
数据的运算
区分



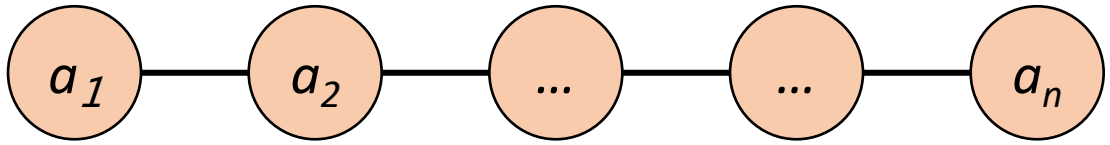
第三节 线性表

一、初识线性表

P285

粉笔

定义：一组特征相同的有限序列，记为L=（a_1，a_2，…，a_n）



特点1：元素个数有限，可以为0

特点2：元素具有顺序性，即有先后次序

特点3：每个元素占相同的存储空间

特点4：除a_1外，其它元素有且仅有一个前驱

特点5：除a_n外，其它元素有且仅有一个后继

二、线性表的顺序存储

（一）顺序表的定义

➢ 线性表的顺序存储又称顺序表

数组下标	序号	顺序表	内存地址
0	1	a_1	$LOC(a_1)$
1	2	a_2	$LOC(a_1)+l$
2	3	a_3	$LOC(a_1)+2\times l$
n-1	n	a_n	$LOC(a_1)+(n-1)\times l$

公式：$LOC(a_i)=LOC(a_1)+(i-1)\times l$

特点：随机访问

- 根据序号可方便找到该元素的位置

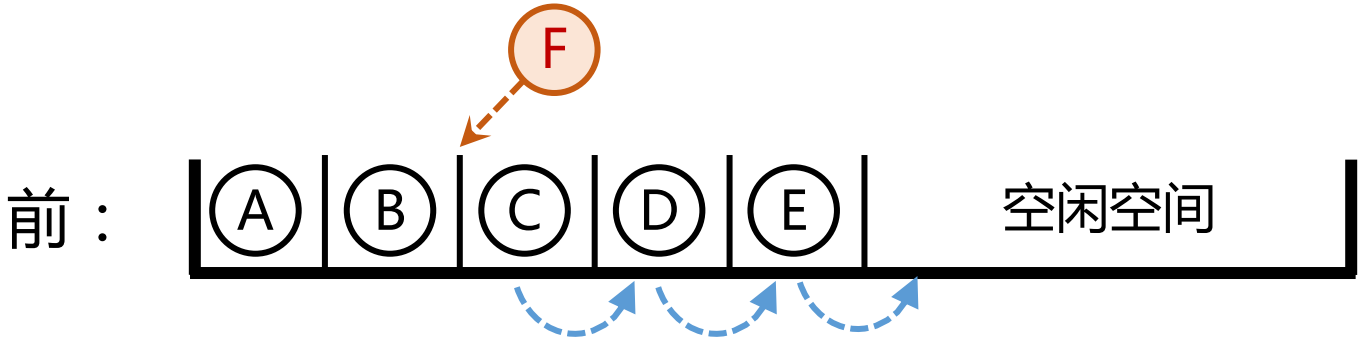
（二）顺序表的基本操作

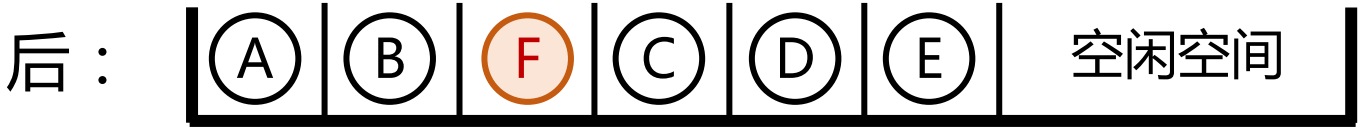
P286

粉笔

操作	说明
1.初始化	构造空表，分配预设大小的空间
2.获取表长	元素的总个数
3.取值（按位查找）	找指定序号位置上的元素
4.查找（按值查找）	找与指定值相等的第一个元素
5.插入	在指定位置插入新元素
6.删除	将指定位置的元素删除

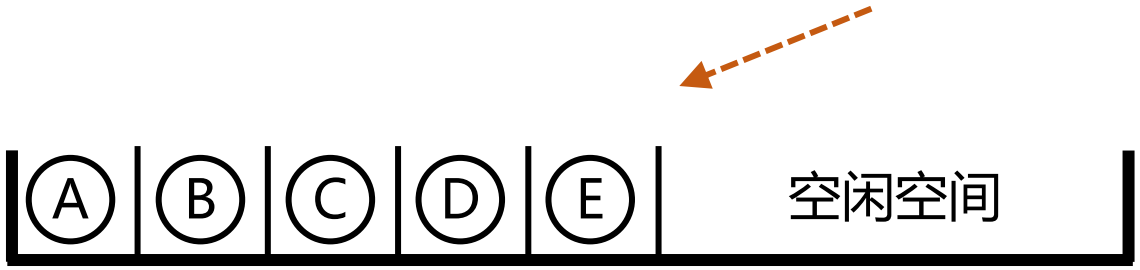
5.插入 （第 i 位置插入新元素）





结论：从最后一个元素到第 i 个元素依次向后移动一个位置，共移动 n-i+1 个元素

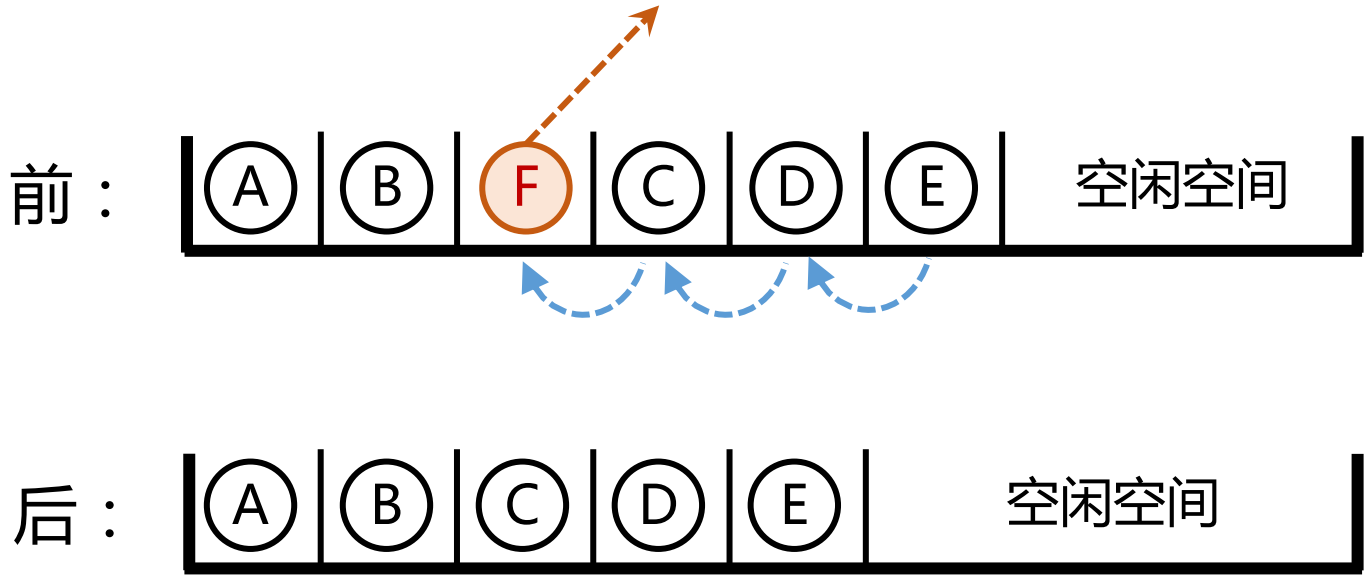
5.插入（第 i 位置插入新元素）



- 最好情况：在表尾插入，移动次数为0
- 最坏情况：在表头插入，移动次数为n
- 平均情况：设p_i是在第i个位置上插入元素的概率，则平均移动次数为：

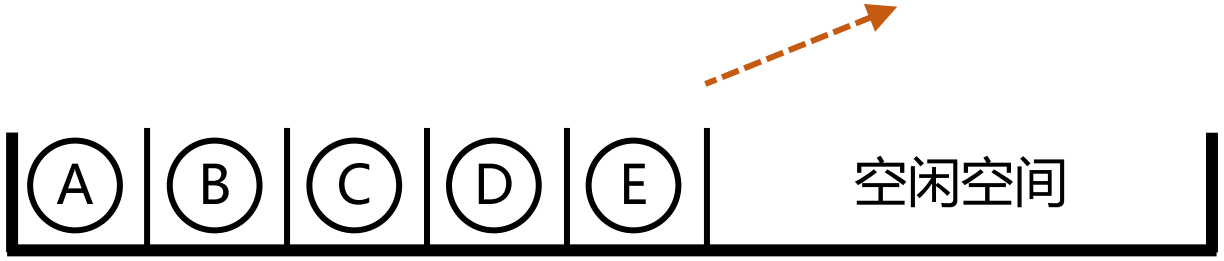
$$\sum_{i=1}^{n+1} p_i\,(n-i+1) = \sum_{i=1}^{n+1} \frac{1}{n+1}(n-i+1) = \frac{n}{2}$$

6.删除 （删除第 i 位置元素）



结论：从第i+1个到最后一个元素依次向前移动一个位置，共移动 n-i 个元素

6.删除（删除第 i 位置元素）



- 最好情况：在表尾删除，移动次数为0
- 最坏情况：在表头删除，移动次数为n-1
- 平均情况：设p_i是在第i个位置上删除元素的概率，则平均移动次数为：

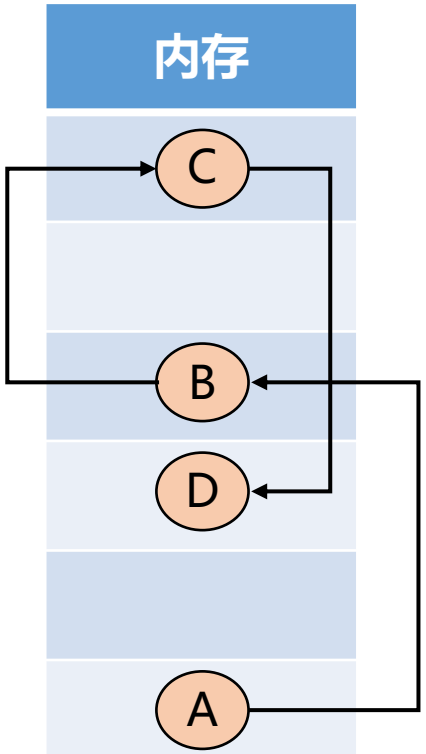
$$\sum_{i=1}^{n} p_i (n-i) = \sum_{i=1}^{n} \frac{1}{n}(n-i) = \frac{n-1}{2}$$

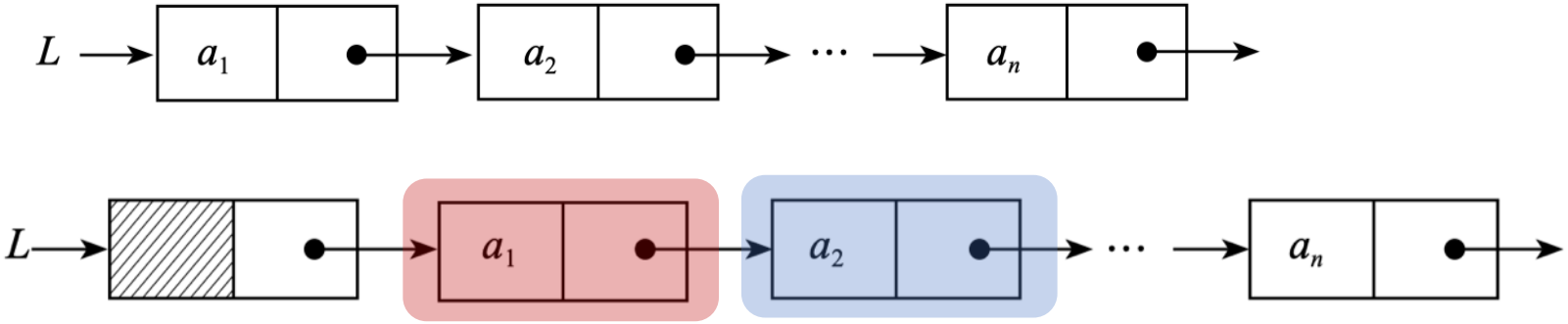
三、线性表的链式存储

P289

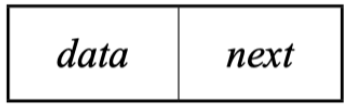
（一）单链表的定义

- 线性表的链式存储又称单链表





1、组成：数据域和指针域



2、本节点的数据：P→data，下一个节点：P→next

3、术语：头指针、头节点、第1个节点

（二）单链表的操作

P290

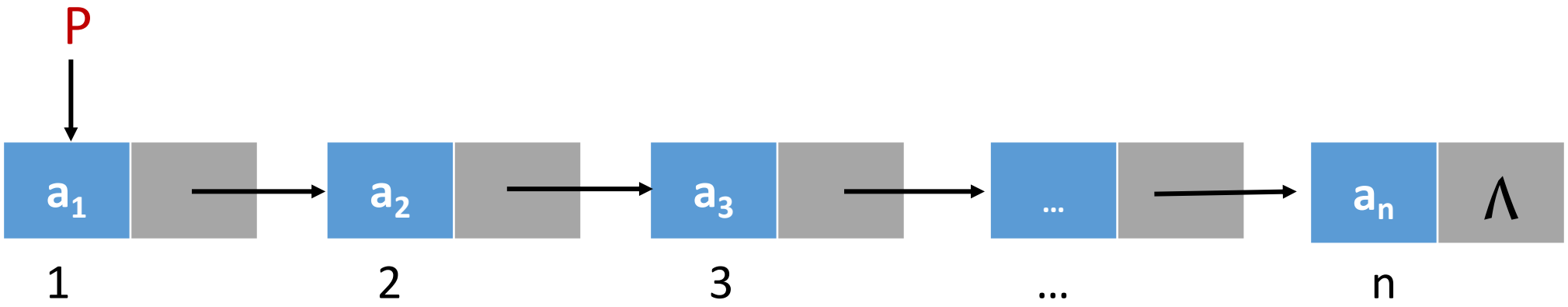
操作	说明
1.初始化	构造空表，只有头节点
2.建立单链表	分头插法和尾插法
3.获取表长	节点的总个数（不含头节点）
4.取值（按位查找）	找指定序号位置上的节点
5.查找（按值查找）	找与指定值相等的第一个节点
6.插入	在指定位置插入新节点
7.删除	将指定位置的节点删除

（二）单链表的操作

P292

粉笔

4.取值（查找第 i 位置的数值） = 5.查找（按值查找）



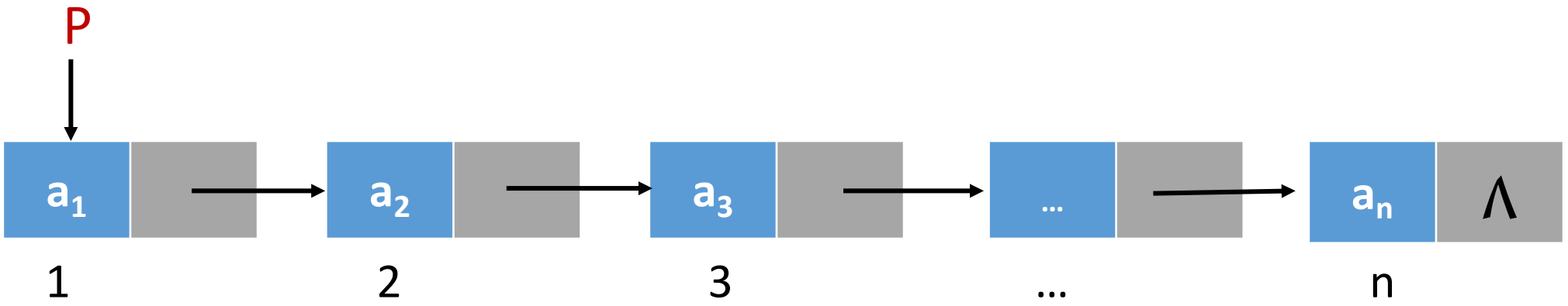
结论：从第一个节点开始，依次向后查找

（二）单链表的操作

P292

粉笔

4.取值（查找第 i 位置的数值） = 5.查找（按值查找）



- 最好情况：查找第1个位置的节点，比较次数为1
- 最坏情况：查找最后位置的节点，比较次数为n
- 平均情况：设p_i是在查找第i个位置上节点的概率，Ci是所需要比较的次数，则平均查找长度为：

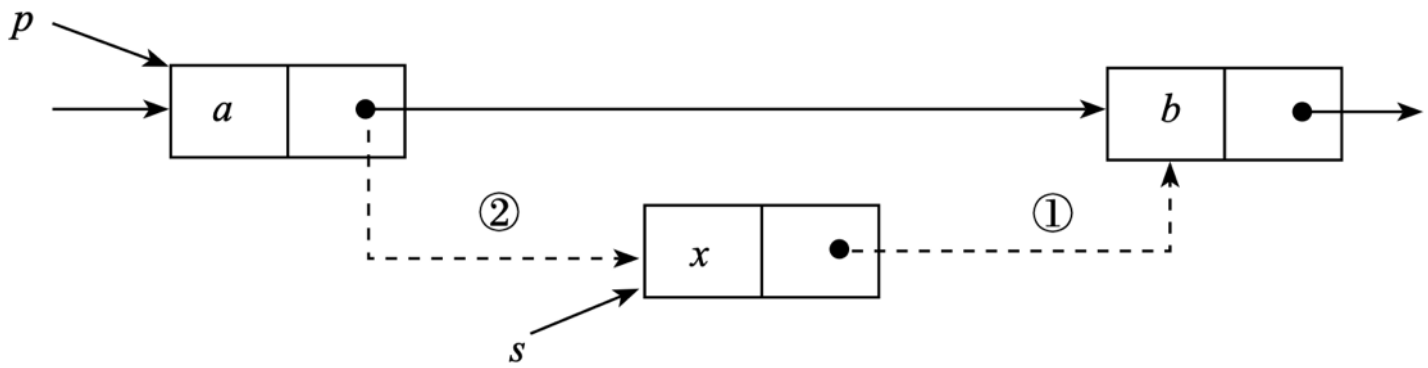
$$\sum_{i=1}^{n} p_i C_I = \sum_{i=1}^{n} \frac{1}{n} \times i = \frac{n+1}{2}$$

（二）单链表的操作

P293

粉笔

6.插入 （若在p节点后面插入一个新节点s）



【补充】

核心步骤（引入q节点）

①s → next = q

②p → next = s

①②顺序可以颠倒

核心步骤（不引入q节点）

①s → next = p → next

②p → next = s

①②顺序不可颠倒

在一个单链表中，已知q所指结点是p所指结点的直接前趋，若在p和q之间插入s结点，这执行（　　）操作。

A.s->next=p->next; p->next=s;

B.q->next=s; s->next=p;

C.p->next=s->next; s->next=p;

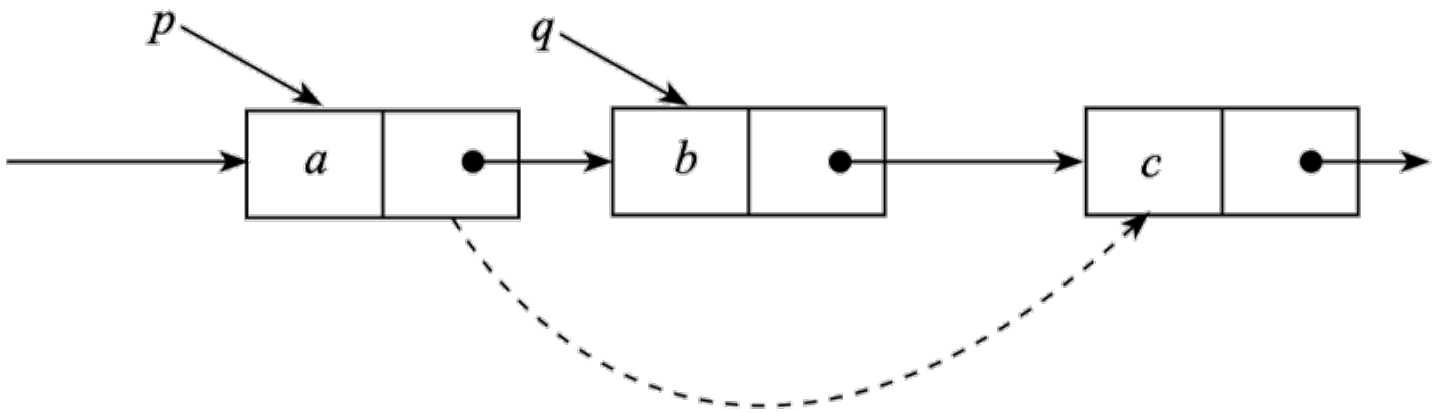
D.p->next=s; s->next=q;

（二）单链表的操作

P294

粉笔

7.删除 （若删除p节点后面的节点）



核心步骤（引入q节点）：

p → next = q → next

【补充】

核心步骤（不引入q节点）：

p → next = p → next → next

顺序表和单链表

总结下

粉笔

	顺序表【顺序存储】	单链表【链式存储】
存储分配	一段连续的空间	一组任意的空间
空间性能	预先分配，容易浪费	不需提前分配
查找	查找方便	查找麻烦
插入和删除	移动大量元素	只需移动指针

关于线性表的顺序存储结构和链式存储结构的描述中，正确的是（　　）。

Ⅰ、线性表的顺序存储结构优于链式存储结构

Ⅱ、链式存储结构比顺序结构能更方便的表示各种逻辑结构

Ⅲ、若频繁使用插入和删除节点操作，则顺序存储结构更优于链式存储结构

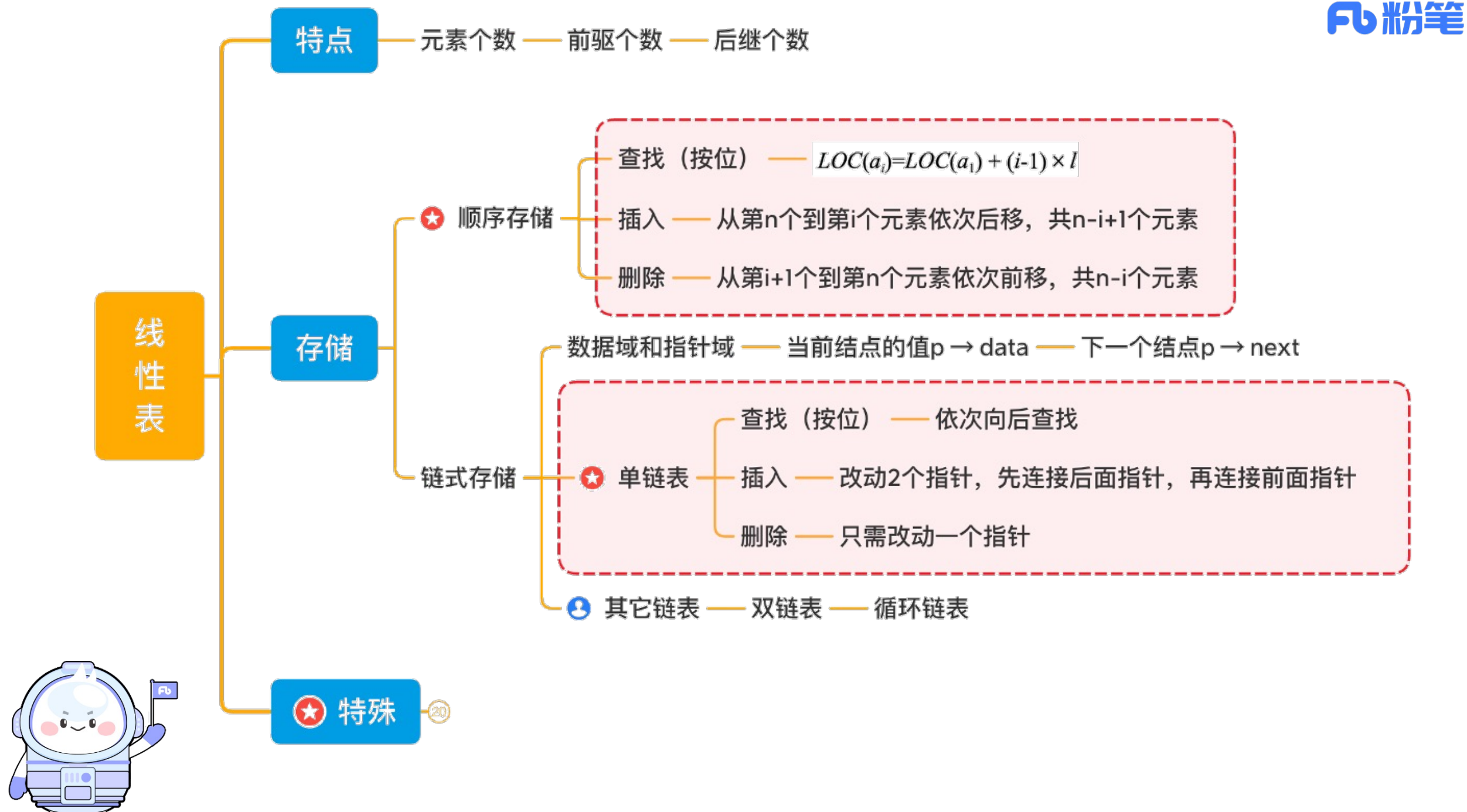
Ⅳ、顺序存储结构和链式存储结构都可以进行顺序查找

A. Ⅰ、 Ⅱ、 Ⅲ

B. Ⅱ、 Ⅳ

C. Ⅱ、 Ⅲ

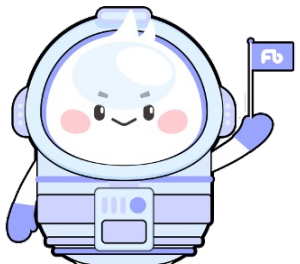
D. Ⅲ、 Ⅳ


线性表
特点
元素个数
前驱个数
后继个数
存储
顺序存储
查找（按位）
$LOC(a_i)=LOC(a_1)+(i-1)\times l$
插入
从第n个到第i个元素依次后移，共n-i+1个元素
删除
从第i+1个到第n个元素依次前移，共n-i个元素
链式存储
数据域和指针域
当前结点的值p → data
下一个结点p → next
单链表
查找（按位）
依次向后查找
插入
改动2个指针，先连接后面指针，再连接前面指针
删除
只需改动一个指针
其它链表
双链表
循环链表
特殊

有疑问没？等你吖

下节内容

P294 ~ P307



粉笔教师

岸上等你

THE TEST

粉笔
教师

光芒万丈
不负理想