## 1.1数据结构的基本概念

数据结构+算法=程序设计

数据（data）：所有能被输入到计算机中，且能被计算机处理的符号的集合。是计算机操作的对象的总称。

数据项（data item）：数据的不可分割的最小单位。一个数据元素可由若干个数据项组成。

数据元素（data element）：由数据项（data item）组成。

关键字（key）：识别元素，主关键字（primary key）唯一识别元素。

数据结构（data structure）：指数据元素之间存在的关系。包含以下三方面：

数据的逻辑结构

数据的存储结构

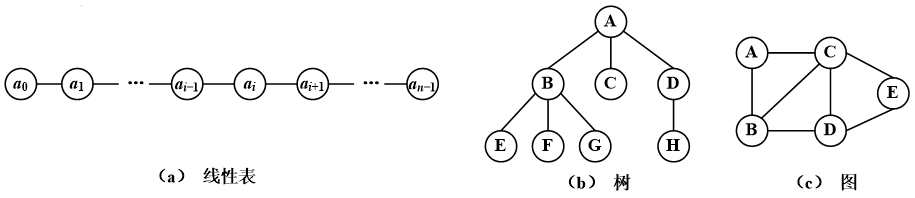
数据操作

## 1.2数据的逻辑结构

1）线性结构：数据元素只有一个前驱数据元素和一个后继数据元素。

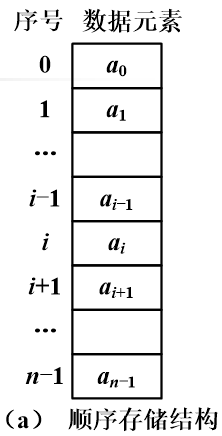
2）树结构：每个数据元素只有一个前驱数据元素，可有零个或若干个后继数据元素。

3）图结构：每个数据元素可有零个或若干个前驱数据元素，零个或若干个后继数据元素。

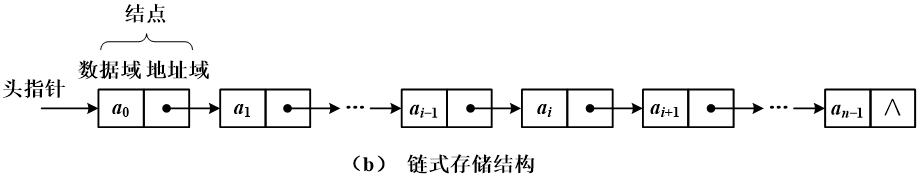


## 1.3数据的存储结构

1）顺序存储结构



2）链式存储结构



## 1.4数据操作

初始化。

判断是否空状态。

存取，指获得、设置指定元素值。

统计数据元素个数。

遍历（traverse），指按照某种次序访问一个数据结构中的所有元素，并且每个数据元素只被访问一次。遍历一种数据结构，将得到一个所有数据元素的线性序列。

插入（insert）、删除（remove）指定元素。

查找（search），指在数据结构中寻找满足给定条件的数据元素。

排列（sort），指对数据元素按照指定关键字值的大小递增（或递减）次序重新排序。

## 1.5数据类型与抽象数据类型

数据类型是指一个类型和定义在这个类型上的操作集合。

抽象数据类型（Abstract Data Type，ADT）是指一个逻辑概念上的类型和这个类型上的操作集合。

一种数据结构的抽象数据类型包括：

数据的逻辑结构

数据操作

（1）集合抽象数据类型：

ADT Set<T> //集合抽象数据类型

{

数据：集合中的数据元素，数据元素的数据类型为T

操作：

boolean isEmpty();

int size();

T search(T key);

boolean contains(T x);

boolean add(T x);

T remove(T key)

void clear();

String toString();

}

（2）实现不同特性的集合：

线性表：表示可重复的无序集合，元素间具有前驱、后继次序关系；不同元素的关键字可重复，采用序号能够识别关键字重复的数据元素。

排序线性表：表示可重复的排序集合，元素按关键字大小次序排序。

散列表：表示不可重复的无序集合，元素关键字不重复，元素键没有次序，不排序。

二叉排序树：表示不可重复的排序集合，元素关键字不重复，元素按关键字什/降序排序。

## 1.6用Java语言的接口描述抽象数据类型

Java语言的接口（interface）是一组抽象方法、常量和内嵌类型的集合。

//集合接口，T是泛型参数

public interface Set<T>

//抽象结合类，没有实现所有抽象方法

public abstract class AbstractSet<T> implement Set<T>

//散列表类接口是引用类型

public class HashSet<T> implement Set<T>

Set<T> set = new HashSet<T>();//接口对象引用实例

set.add(x);//运行时多态，执行HashSet<T>类实现的add(x)方法

## 2.1什么是算法

算法（Algorithm）是指解题方案的准确而完整的描述。即是一组严谨地定义运算顺序的规则，并且每一个规则都是有效的，且是明确的，没有二义性，同时该规则将在有限次运算后可终止。

算法定义：

1. 有穷性
2. 确定性
3. 输入
4. 输出

算法设计目标：

1. 正确性
2. 可读性
3. 健壮性
4. 高时间效率
5. 高空间效率

算法描述：

//在当前数据结构中，顺序查找与key相等的元素（数据类型为T）；

//key提供查找条件的关键字元素

search(T key)

{

for(elem : 数据结构中的每个元素)

if(key与elem元素相等) {

//由T类型约定两个元素相等的比较规则查找成功，返回元素或元素位置；

查找成功，返回元素或元素位置；

}

查找不成功，返回查找不成功标记；

}