**什么是算法**

一个**算法**（Algorithm）是一个有穷规则的集合，其规则确定一个解决某一特定类型问题的操作序列。

1. 算法定义：
2. 有穷性
3. 确定性
4. 输入
5. 输出
6. 可行性
7. 算法设计目标：
8. 正确性
9. 可读性
10. 健壮性
11. 高时间效率
12. 高空间效率
13. 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

元素**search**(T key)

{

for(elem:数据结构中的每个元素) //遍历

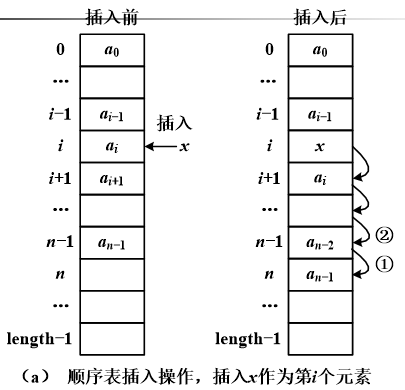
if(key与elem元素相等)

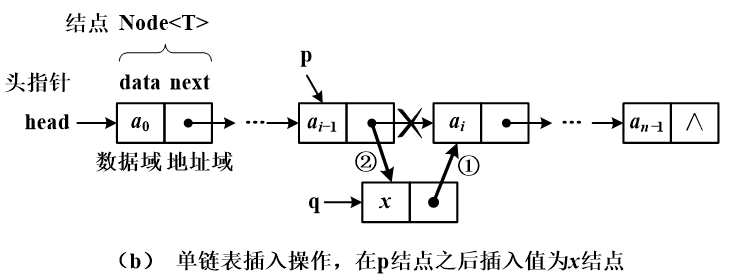
//由T类型约定两个元素相等的比较规则查找成功，返回元素或元素位置；

//查找不成功，返回查找不成功标记；

}

**线性表插入操作：**





**线性存储的随机数组合**

1. 使用Java的一维数组，对象数组作为方法的参数和返回值；
2. 输出对象的通用方法；
3. 声明类，包含对数组操作的通用方法；
4. 随机数集合，线性存储。
5. Java的数组

Java数组的特点是动态数组、具有长度属性length、引用数据类型。

1. for语句的逐元循环（for each loop），作用于数组的语法格式如下：

for(类型 变量:数组)

1. 声明Array1类

public class Array1 {

public static void print(Object[] value)

public static Integer[] randomInteger(int n, int size)

}

**注意：java.lang.Integer是int整数类型的包装类**

调用时默认将int整数与Integer对象相互转换。例如：

Integer key = new Integer(100);

Integer key = 100;

//Java自动将int整数封装成Integer

int i = key.intValue();

int i = key;

//Java自动调用Integer的intValue()方法，将Integer对象转换成int整数

**随机数集合**

Integer[] randomDifferent(int n,int size)

//返回n个互异的随机数，范围是0～size-1

Integer[] randomSorted(int n,int size)

//返回n个排序的随机数

Integer[] randomDifferentSorted(int n,int size)

//返回n个互异的排序的随机数