程序设计 = 数据结构 + 算法

数据结构就是关系，就是数据元素相互之间存在得一种或多种特定关系得集合。

逻辑结构：

集合结构：其中得数据元素除了同属于一个集合外，它们之间没有其他的关系。

线性结构：其中的数据元素是一对一的关系。

树形结构：其中的数据元素之间存在一种一对多的层次关系。

图形结构：其中的数据元素是多对多的关系。

物理结构：

顺序存储：把数据元素存放在地址连续的存储单元里，其数据间的逻辑关系和物理关系是一致的。

链式存储：把数据元素存放在任意的存储单元里，这组存储单元可以是连续的，也可以是不连续的。其数据元素存储关系并不能反映其逻辑关系，需要存放指针，其指向下一个元素的地址。

算法的五个基本特征：输入、输出、有穷性、确定性、可行性。

正确性分以下四个层次：

1. 算法程序没有语法错误；
2. 算法程序对于合法输入能够产生满足要求的输出；
3. 算法程序对于非法输入能够产生满足规格的说明；
4. 算法程序对于故意刁难的测试输入都有满足要求的输出结果。
5. **int** i = 1, n = 100;
6. **while** (i < n)
7. i \*= 2

I = 1 2 4 8 16，当时就退出循环，即，所以其时间复杂度为。

线性表

像排队一样，具有线一样性质的结构。

1. 它是一个序列，元素之间是有个先来后到的。
2. 若元素存在多个，则第一个元素无前驱，而最后一个元素无后继，其他元素都有且只有一个前驱和后继。
3. 所处理的元素都是有限的。

线性表元素的个数n (n>=0)定义为线性表的长度，当n=0时，称为空表。

Operation

InitList(\*L)：初始化操作，建立一个空的线性表L。

ListEmpty(L)：判断线性表是否为空表，若线性表为空，返回true，否则返回false。

ClearList(\*L)：将线性表清空。

GetElem(L, i, \*e)：将线性表L中的第i（i从1开始）个位置元素值返回。

LocateElem(L, e)：在线性表L中查找与给定值e相等的元素，如果查找成功，返回该元素在表中序号表示成功；否则，返回0表示失败。

ListInsert(\*L, i, e)：在线性表L中第i个位置插入新元素e。

1. 若插入位置不合理，抛出异常；
2. 若线性表长度大于等于数组长度，则抛出异常或动态增加数组容量。
3. 从最后一个元素开始向前遍历到第i个位置，分别将它们都向后移动一个位置。
4. 将要插入元素填入位置i处；
5. 线性表长+1

ListDelete(\*L, i, \*e)：删除线性表L中第i个位置元素，并用e返回其值。

ListLength(L)：返回线性表L的元素个数。

Q1：实现两个线性表A、B的并集操作，使得集合。- P7

线性表的顺序存储结构：

1. #define MAXSIZE 20
2. **typedef** **int** ElemType;
3. **typedef** **struct**
4. {
5. ElemType data[MAXSIZE];
6. **int** length;
7. } SqList;

需要的三个属性：

1. 存储空间的起始位置，数组data；
2. 线性表的最大存储容量MAXSIZE；
3. 线性表的当前长度length。

P09