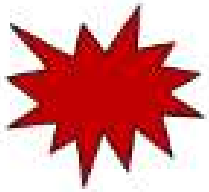
数据结构与算法



我遇到了一个难题！

疫情期间，
如何快速的创建学生
健康情况表

1

投票 最多可选1项

你认为最喜欢的方法？

A 传统的方法

B 信息化的手段

C 靠自己智慧的大脑

马小龙是1905211班10号的帅哥一枚，勤劳、爱学、努力向上；
湖北武汉，健康。

李舒凡是1905213班21号的美女一枚，好学、活泼、会唱歌；
湖南长沙，健康。

姓　名	学　　号	性　别	年龄	健康情况
马小龙	190521110	男	18	健康
李小亨	190521101	男	21	健康
……	……	……	……	……
王小美	190521140	女	18	感染肺炎

清晰明了，一目了然！

我们想实现的

单选题 5分

实现这个过程的第一步？

A 对问题进行分析，明确数据特性

B 选择合适的开发工具

C 选择一门高级开发语言

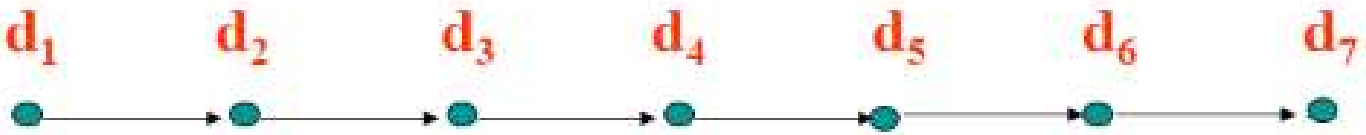
D 其他

我们会选用哪种结构呢？

线性结构特点：　　一对一的关系

在非空的线性表中：

- 1）有且仅有一个开始结点d_1，它没有直接前趋.
- 2）有且仅有一个终端结点d_n，它没有直接后继.
- 3）其余的内部结点d_i $(2\le i\le n-1)$都有且仅有一个直接前趋d_{i-1}和一个直接后继d_{i+1}。



线性结构

我们的线性表

线性表是具有相同数据类型的n（n≥0）个数据元素的有限序列，通常记为：

$(a_1, a_2, \ldots a_{i-1}, a_i, a_{i+1}, \ldots a_n)$

其中n为表长， n＝0 时称为空表。

在线性表中相邻元素之间存在着顺序关系。对于元素a_i 而言， a_{i-1} 称为 a_i 的直接前趋，a_{i+1} 称为 a_i 的直接后继。

8

我们的线性表

（1）简单的线性表

例如一年12个月：

（1，2，3，4，5，6，7，8，9，10，11，12）

在C或C+ +语言中我们可以把它们定义为数值型。

又例如26个英文字母表：

（a，b，c，d，e，f，g，……，x，y，z）

在C或C+ +语言中我们可以把它们定义为字符型。

9

我们的线性表

（2）复杂的线性表

例如绪论中引用的学生入学情况表（表1-1）

学号	姓名	性别	入学总分
01	丁一	男	540
02	马二	男	535
03	张三	女	538
04	李四	男	530
05	王五	女	545
06	赵六	男	528
07	钱七	女	532
08	孙八	男	537
09	冯九	女	526
10	郑十	女	535

10

我们的线性表

一元多项式的运算

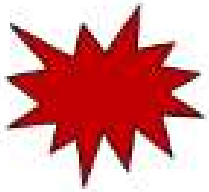
稀疏多项式的运算

线性结构

图书信息管理系统

数据元素间具有线性结构特征的实际应用问题。

11



最终我们找到了方法

——线性表！

我们的线性表

下面的工作需要我们一起完成！

1.线性结构的抽象数据类型；

2.不同存储结构下的线性表如何区别。

13

单选题 2分

下列哪种说法是不正确的？

A 线性表是具有相同数据类型的数据元素集合。

B **线性表中的数据元素是有限的。**

C **线性表中的数据元素是有顺序的。**

D 线性表的表长n是一个确定的值，不发生变化。

线性表的抽象数据类型定义

抽象数据类型(Abstract Data Type 简称ADT)是指一个数学模型以及定义在此数学模型上的一组操作。

抽象数据类型需要通过固有数据类型(高级编程语言中已实现的数据类型)来实现。

抽象数据类型是与表示无关的数据类型，是一个数据模型及定义在该模型上的一组运算。

对一个抽象数据类型进行定义时，必须给出它的名字及各运算的运算符名，即函数名，并且规定这些函数的参数性质。

一旦定义了一个抽象数据类型及具体实现，程序设计中就可以像使用基本数据类型那样，十分方便地使用抽象数据类型。

线性表是最基本、最简单、也是最常用的一种数据结构。

线性表中数据元素之间的关系是一对一的关系，即除了第一个和最后一个数据元素之外，其它数据元素都是首尾相接的(注意，这句话只适用大部分线性表，而不是全部。比如，循环链表逻辑层次上也是一种线性表(存储层次上属于链式存储)，但是把最后一个数据元素的尾指针指向了首位结点)。

我们说"线性"和"非线性"，只在逻辑层次上讨论，而不考虑存储层次，所以双向链表和循环链表依旧是线性表。

在数据结构逻辑层次上细分，线性表可分为一般线性表和受限线性表。一般线性表也就是我们通常所说的"线性表"，可以自由的删除或添加结点。受限线性表主要包括栈和队列，受限表示对结点的操作受限制。

线性表的逻辑结构简单，便于实现和操作。因此，线性表这种数据结构在实际应用中是广泛采用的一种数据结构。

线性表的抽象数据类型ADT

ADT LIST{

数据对象：D= { a_i | $a_i \in$ Elemset ,i=1,2,...n,n>=0 }

数据关系：R1= { < a_{i-1},a_i > | a_{i-1}, $a_i \in$ D i=2,...,n }

基本操作：

（1）InitList(&L)；

（2）DestroyList(&L)；

（3）ClearList(&L);

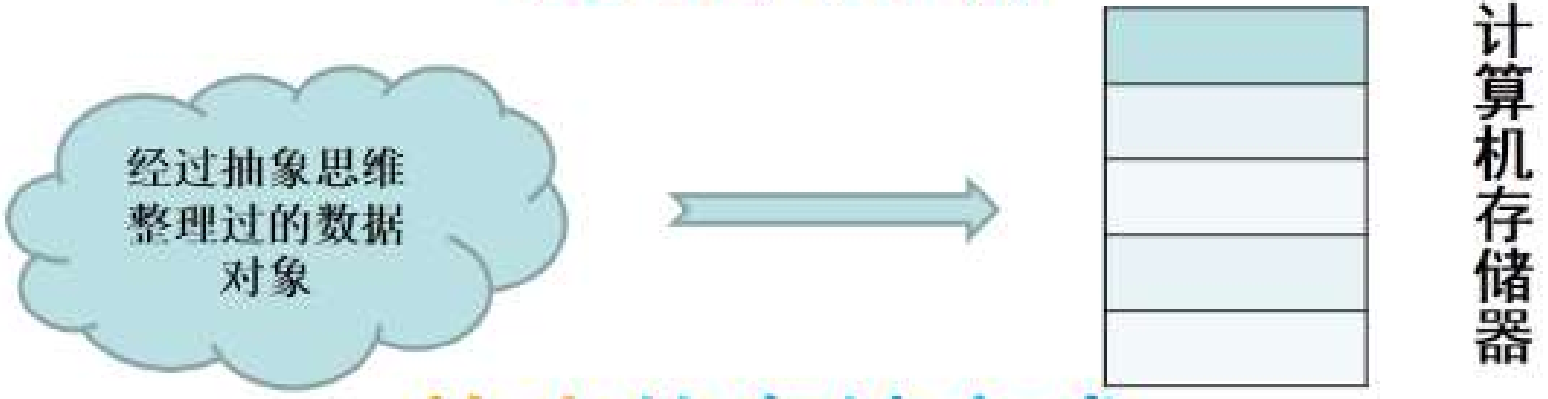
.....

}ADT LIST

接下来，我们要做什么？

实现逻辑思维到物理存储的

完美映射



恰当的存储方式

不同存储结构下的线性表如何区别

优缺点

① 顺序存储时，相邻数据元素的存放地址也相邻（逻辑与物理统一）；要求内存中可用存储单元的地址必须是连续的。
优点：存储密度大（=1），存储空间利用率高。缺点：插入或删除元素时不方便。
②链式存储时，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针
优点：插入或删除元素时很方便，使用灵活。缺点：存储密度小（<1），存储空间利用率低。

使用情况

顺序表适宜于做查找这样的静态操作；链表宜于做插入、删除这样的动态操作。
若线性表的长度变化不大，且其主要操作是查找，则采用顺序表；
若线性表的长度变化较大，且其主要操作是插入、删除操作，则采用链表。

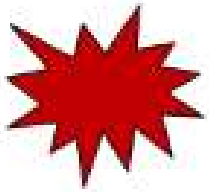
两者比较

基于空间的比较

1.存储分配的方式
顺序表的存储空间是静态分配的
链表的存储空间是动态分配的
2.存储密度 = 结点数据本身所占的存储量/结点结构所占的存储总量
顺序表的存储密度 = 1
链表的存储密度 < 1

基于时间的比较

1.存取方式
顺序表可以随机存取，也可以顺序存取
链表是顺序存取的
2.插入/删除时移动元素个数
顺序表平均需要移动近一半元素
链表不需要移动元素，只需要修改指针



接下来，我们要开始了解

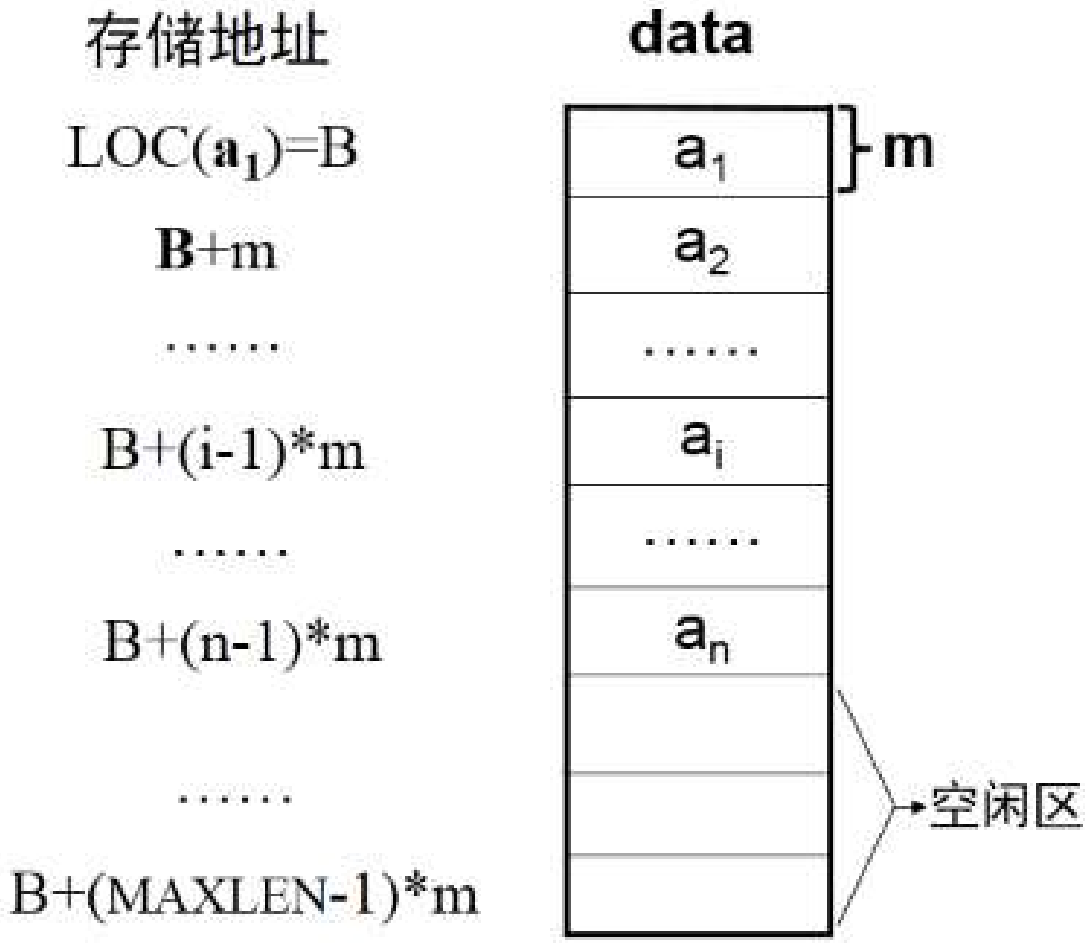
线性表的顺序表示与实现

23

线性表的顺序存储是指在用一组地址连续的存储单元依次存储线性表的数据元素。

我们把用这种存储形式存储的线性表称为**顺序表**。

24


存储地址
data
LOC(a1)=B
B+m
……
B+(i-1)*m
……
B+(n-1)*m
……
B+(MAXLEN-1)*m
a1
a2
……
ai
……
an
m
空闲区

设a_1的存储地址$LOC(a_1)$为首地址B，每个数据元素占m个存储单元，则第i个数据元素的地址为：

$$LOC(a_i)=LOC(a_1)+(i-1)*m$$

即：

$$LOC(a_i)=B+(i-1)*m \qquad 1<=i<=n$$

顺序表具有按数据元素的序号随机存取的特点。

26

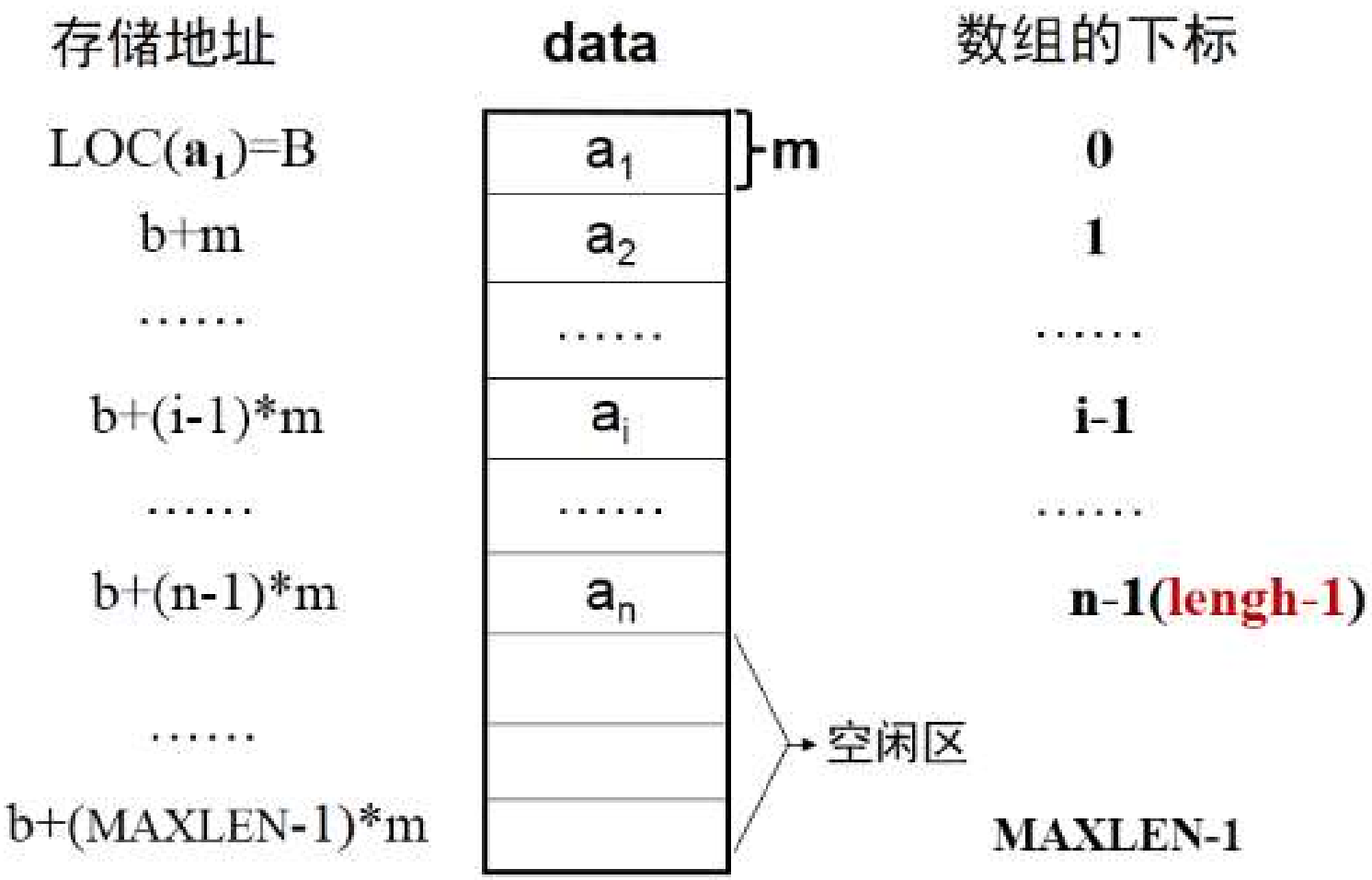
在程序设计语言中，一维数组在内存中占用的存储空间就是一组连续的存储区域，可以用一维数组来表示：

```
datatype  data[MAXLEN];
int  lengh;
```

设表长为lengh，则数据元素分别存放在data[0]到data[lengh-1]中。

27

线性表的顺序存储示意图



多选题 3分

关于顺序表的特点，下述内容哪些正确？

A 在顺序表中，各个表项的逻辑顺序与其存放的物理顺序是一致的，即第i个表项存储于第i个物理位置（1<=i<=n）。

B **在顺序表中，各个表项的逻辑顺序与物理顺序无关。**

C 对顺序表中所有表项，既可以进行顺序访问，也可以进行随机访问。

D 顺序表仅能按照表项的序号直接访问表项。

顺序表的基本运算

```
# define LIST_INIT_SIZE  100  // 线性表存储空间的初始分配量
# define LISTINCREMENT  10  // 线性表存储空间的分配增量
typedef struct {
ElemType  *elem;  // 存储空间基址
int length;          //当前长度
int listsize;          //当前分配的存储容量（以sizeof(ElemType)为单位）
}Sqlist;
```

顺序表的基本运算

1.顺序表的初始化即构造一个空表，将L设为指针参数，动态分配存储空间，将线性表的当前长度lengh设为0。算法如下：

```
void InitList_Sq(SqList &L) {  //构造一个空的顺序表
  L.elem=（ElemType*）
        malloc(LIST_INIT_SIZE*sizeof(ElemType);
        if(!L.elem) Error(“Overflow!”);   //存储分配失败
  L.length=0;                                    //空表长度为
  L.listsize=LIST_INIT_SIZE;//初始存储容量
}// InitList_Sq
```