数据结构与算法

疫情面前，让我们一起努力！

1

之前的课程，大家一起学习了！

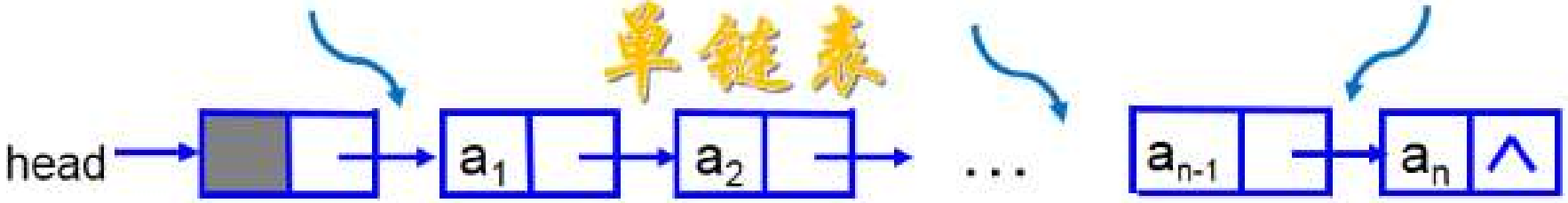
线性表的**顺序**表示与实现

线性表的**链式**表示与实现

线性表的插入、删除操作？

顺序表

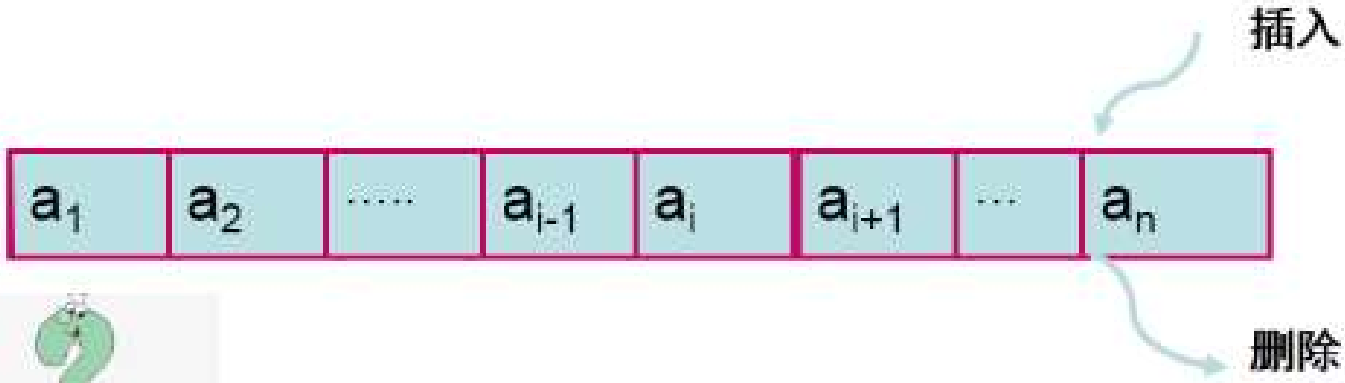
单链表



如果我们将插入、删除位置限定？

3

设想1：将限制在表的一端进行插入和删除运算的线性表。



会产生怎样的效果？

栈的定义

栈是限制在表的一端进行插入和删除运算的线性表。

通常称插入、删除的这一端为栈顶(Top)，另一端为栈底(Bottom)。

当栈中没有元素时称为空栈。

5

单选题 1分

假设栈S=(a_1，a_2，a_3，...a_n)，则a_1称为栈底元素，a_n为栈顶元素。栈中元素按a_1，a_2，a_3，...a_n的次序进栈，退栈的第一个元素应为

A a_1

B a_n

C a_{n-1}

D a_2

栈的定义

上述过程中，不难发现：

最先入栈的元素，最后出栈；

最后入栈的元素，最先出栈。

换句话说，栈的修改是按后进先出的原则进行的。

7

抽象数据类型——栈

ADT Stack{

数据对象：

D={a_i | a_i∈ ElemSet, i=1,2,…,n (n>=0)}

数据关系：R={<a_{i-1},a_i>| a_{i-1},a_i∈ D,i=2,3,…,n}

约定a_n端为栈顶，a_1端为栈底。

基本操作：

InitStack(&S)

操作结果：建立一个空栈S

DestroyStack(&S)

操作结果：栈S被销毁

抽象数据类型——栈

ClearStack(&S)　操作结果：将S栈清空

StackEmpty(S　操作结果：若栈S为空栈，返回true,否则返回false

StackLength(S)　操作结果：返回S的元素个数

GetTop(S,&e) //读栈顶元素　操作结果：用e返回栈顶元素

Push(&S,e) //入栈　操作结果：将元素e插入到栈顶

Pop(&S,&e) //出栈　操作结果：删除S的栈顶元素，用e返回

}ADT Stack

栈的表示和实现

由于栈的逻辑结构与线性表相同，因此线性表的存储结构对栈也适应。

- 顺序栈
- 链栈

栈的存储结构

顺序存储结构： 利用一组地址连续的存储单元依次存放自栈底到栈顶的数据元素。

顺序栈

链式存储结构： 用于收集计算机存储器中所有空闲存储空间，来保存自栈底到栈顶的数据元素。

链栈

多选题 2分

在确定使用顺序存储结构后，该从哪几个方面描述顺序栈？

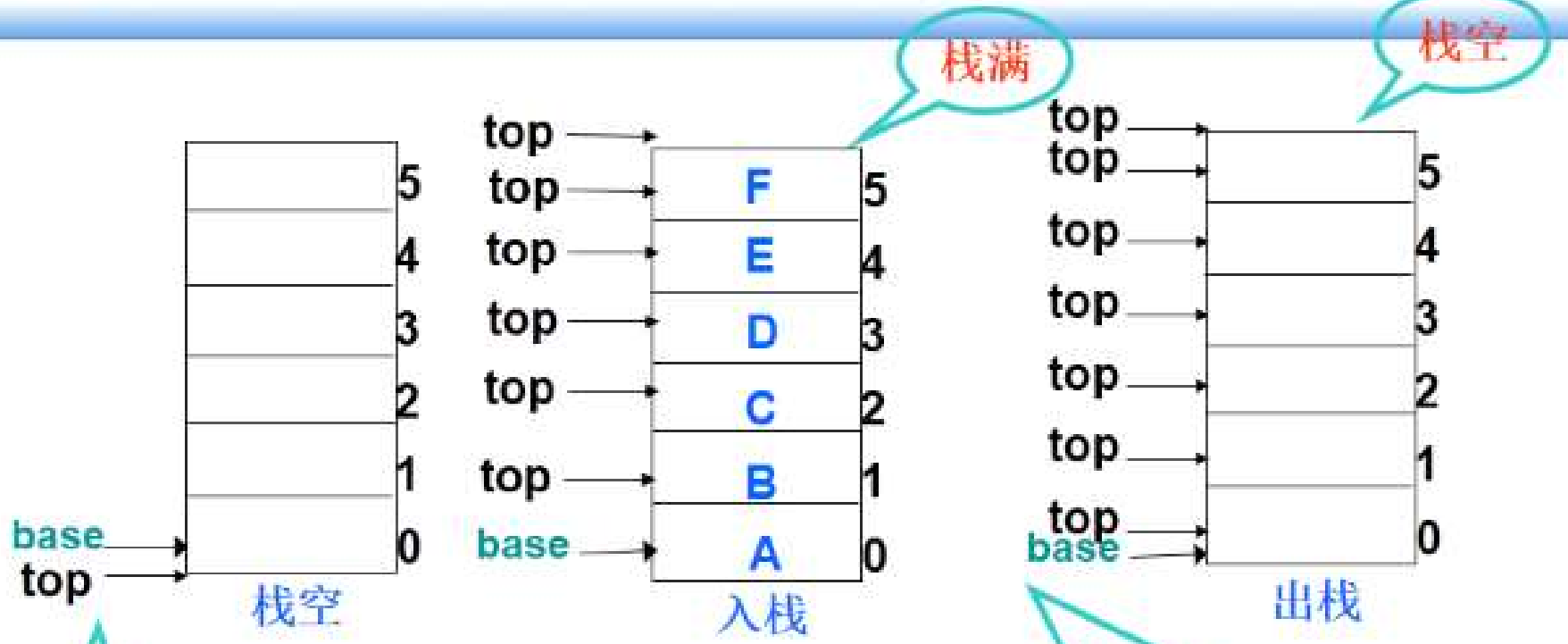
A 定义一个指向栈底的指针；

B 定义一个指向栈顶的指针；

C 定义一个指向基地址的指针；

D 不需要定义指针。

顺序栈的入栈与出栈



栈顶指针top,指向栈底位置

设数组大小为stacksize
top==base,栈空，此时出栈，则下溢（underflow)
top== stacksize,栈满，此时入栈，则上溢（overflow)

顺序表的基本运算

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
```

顺序栈的表示和实现

1、构造一个空栈

```
（1）Status InitSatck(SqStack  &S) {
（2）
S.base=(SElemType*)malloc(STACK_INIT_SIZE
*sizeof(SElemType));
（3）if(!S.base) exit(OVERFLOW);
（4） S.top=S.base;
（5） S.stacksize= STACK_INIT_SIZE;
}//end InitSatck
```

顺序栈的表示和实现

2、返回栈顶元素

```
（1） Status GetTop(SqStack  S，SElemTtype  &e)
{
（2）  if (S.top==S.base) return ERROR;
（3）       e=*(S.top-1);
（4）       return OK;
   }//end  GetTop
```

顺序栈的表示和实现

3. 入栈

进栈运算是在栈顶位置插入一个新元素x，其算法步骤为：

(1) 判栈是否为满，若栈满，作溢出处理，并返回0；

(2) 若栈未满，栈顶指针top加1；

(3) 将新元素x送入栈顶，并返回1。

顺序栈的表示和实现

3、入栈

```
（1）Status Push(SqStack &S，SElemType e) {
（2）if(S.top-S.base>=S.stacksize){
（3）S.base=(SElemType*)realloc(S.base,(S. stacksize
+ STACKINCREMENT)* sizeof(SElemType));
（4） if(!S.base) exit(OVERFLOW);
（5） S.top=S.base+S.stacksize;
（6） S.stacksize+= STACKINCREMENT; }
（7） *S.top++=e;
} //end Push
```

顺序栈的表示和实现

4. 出栈

出栈运算是指取出栈顶元素，赋给某一个指定变量x，其算法步骤为：

(1) 判栈是否为空，若栈空，作下溢处理，并返回0；

(2) 若栈非空，将栈顶元素赋给变量x；

(3) 指针top减1，并返回1。

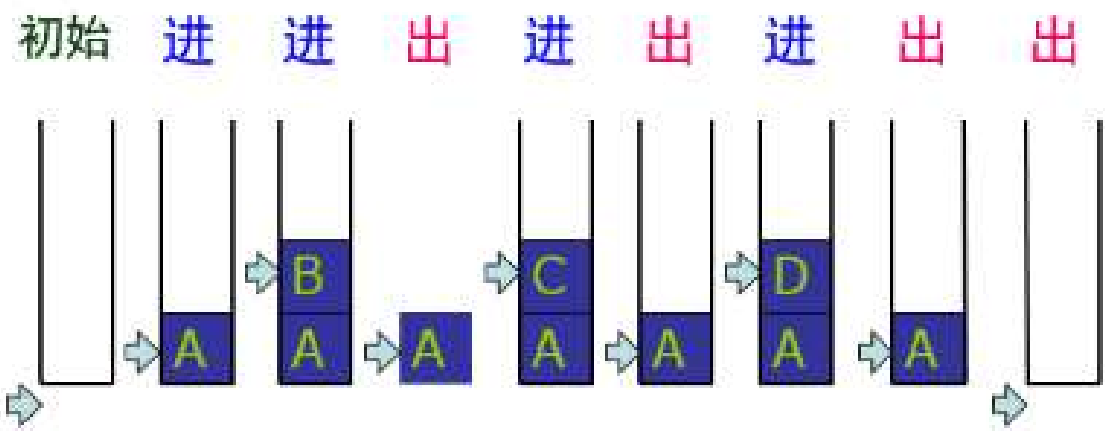
顺序栈的表示和实现

4、出栈

```
（1）Status Pop(SqStack &S，SElemType &e)
{
 （2） if(S.top==S.base) return ERROR;
 （3）    e=*--S.top;
 （4）    return OK;
}//end Pop
```

单选题 2分

思考：A,B,C,D按此顺序依次进栈



出栈元素顺序可能是： B → C → D → A吗？

A 可能

B 不可能

栈的定义-总结

上述过程中，我们可以总结：

栈中增加元素时，Top++；

栈中减少元素时，Top--。

换句话说，栈的基础操作的关键是Top指针。

栈的特性：后进先出

如何在实际问题中，运用栈的特性。