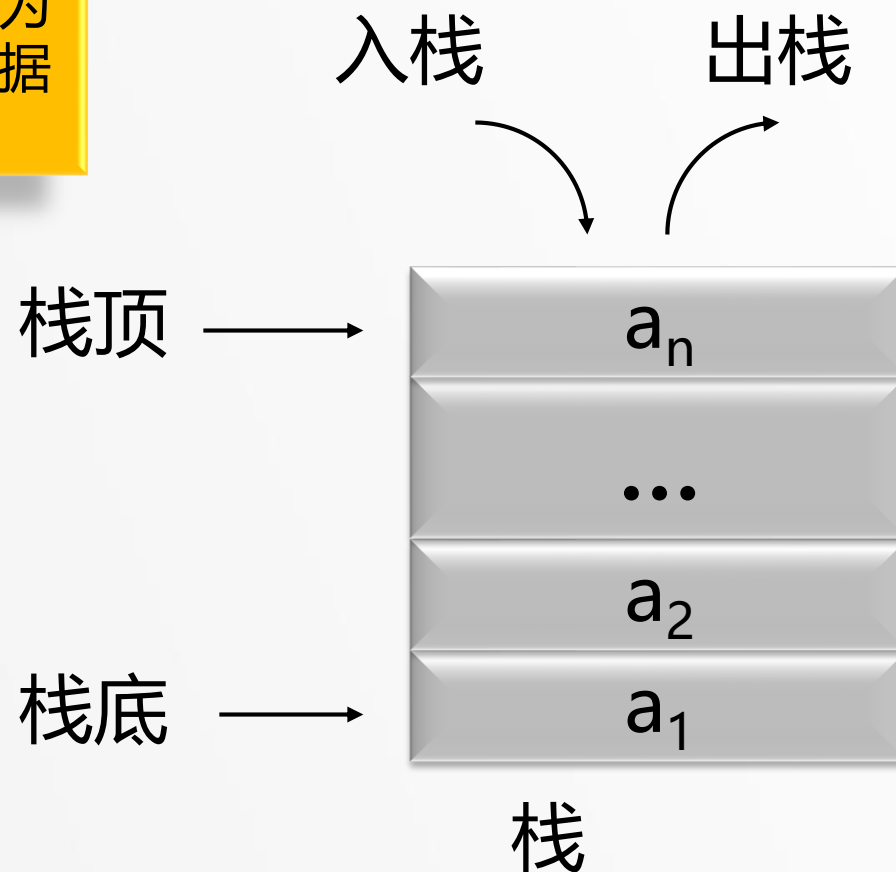


2.2.1 栈的定义

栈是限定在表的同一端进行插入或删除操作的线性表。进行插入或删除操作的一端称为**栈顶**，另一端称为**栈底**。没有数据元素的栈称为**空栈**。插入数据元素的操作称为**入栈**，删除数据元素的操作称为**出栈**。

栈的运算特性：
后进先出 (Last In First Out--**LIFO**)
或先进后出 (First In Last Out--**FILO**)



- 栈的抽象数据类型定义

ADT Stack {

数据对象:

$D = \{ a_i \mid a_i \in \text{ElemSet}, i=1,2,\dots,n, n \geq 0 \}$

数据关系:

$R1 = \{ \langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D, i=2,\dots,n \}$

约定 a_n 端为栈顶, a_1 端为栈底

基本操作:

} ADT Stack

Stack_Init (StackPtr s)

Stack_Destroy (StackPtr s)

Length_Stack (StackPtr s)

Stack_Empty(StackPtr s)

Stack_Clear (StackPtr s)

Stack_Top(StackPtr s, StackEntry *item)

Stack_Push(StackPtr s, StackEntry item)

Stack_Pop(StackPtr s, StackEntry *item)

.....

测试栈的基本操作

序号	名称	功能	栈的基本操作函数
1	Test_PushPrint	建立空栈，然后读入待入栈的数据进行入栈；打印栈中数据看是否和入栈数据一致	Stack_Init, Stack_Push, Stack_Print
2	Test_PushPop	建立空栈，然后读入待入栈的数据进行入栈；将所有数据出栈，看是否与入栈数据一致	Stack_Init, Stack_Push, Stack_Pop
3	Test_TopSizeClearEmpty	建立空栈，然后读入待入栈的数据,查看栈顶元素；求栈的长度；清空栈;查看栈是否已经清空	Stack_Init, Stack_Push, Stack_Top, Stack_size, Stack_Clear, Stack_Empty

• 讨论1:

- 采用顺序存储结构实现的线性表还是链式存储结构的线性表为基础，编程实现栈的基本操作？
- 如果采用顺序存储结构的线性表，栈顶在哪里？如果采用带头结点的单链表为存储结构，栈顶在哪里？为什么？

• 讨论2

- 如果不采用线性表的基本操作为基础，我们自己去完整的编程实现栈的基本功能，采用哪种存储结构更合适？
- 需要做哪些工作？
- 什么时候需要我们自己这样编程？