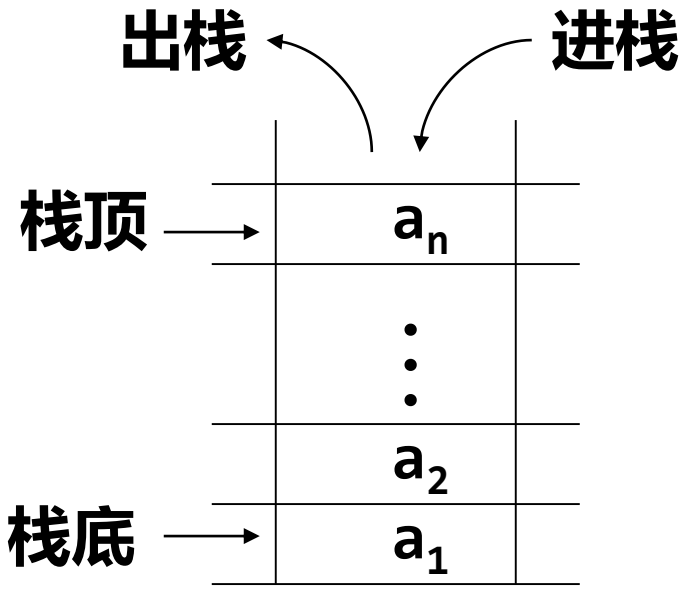
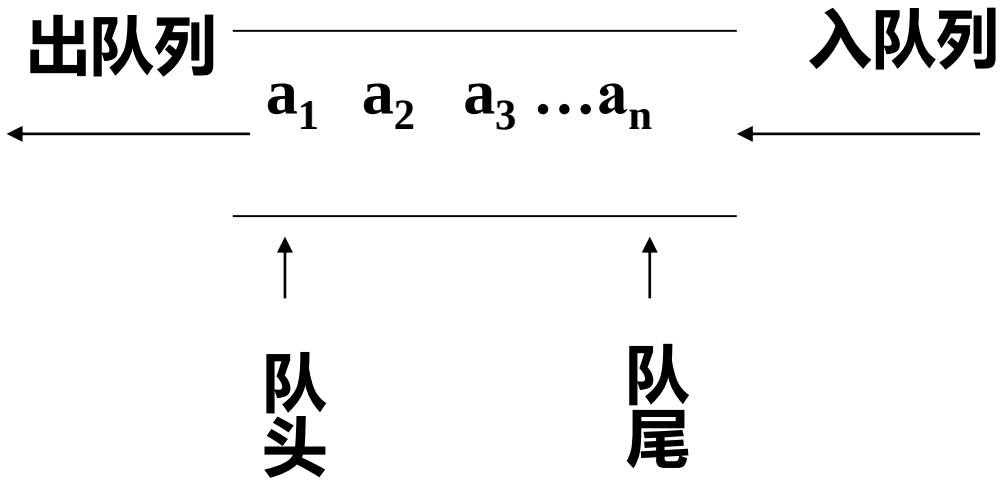
3.1 栈和队列的概念

- 栈和队列是操作受限的线性表
- 栈限定在表尾进行插入或删除
  表尾端称栈顶，表头端称栈底
- 栈的特点：后进先出（LIFO）



- 队列限定在一端进行插入、一端删除
  插入端称队尾，删除端称队头
- 队列的特点：先进先出（FIFO）



栈和队列的抽象数据类型

ADT Stack is

Operations

Stack SetNullStack(void)

创建一个空栈

int IsNullStack(Stack stack)

判断栈是否为空栈

void Push(Stack stack,Datatype x

往栈中插入（或称推入）一个元素

void Pop(Stack stack)

从栈中删除（或称弹出）一个元素

Datatype Top (Stack stack)

求栈顶元素的值

End ADT Stack

ADT Queue is

operations

Queue SetNullQueue (void)

创建一个空队列

int IsNullQueue (Queue que)

判断队列que是否为空

void enQueue (Queue que,Datatype x)

向队列que中中插入元素x

void DeQueue (Queue que)

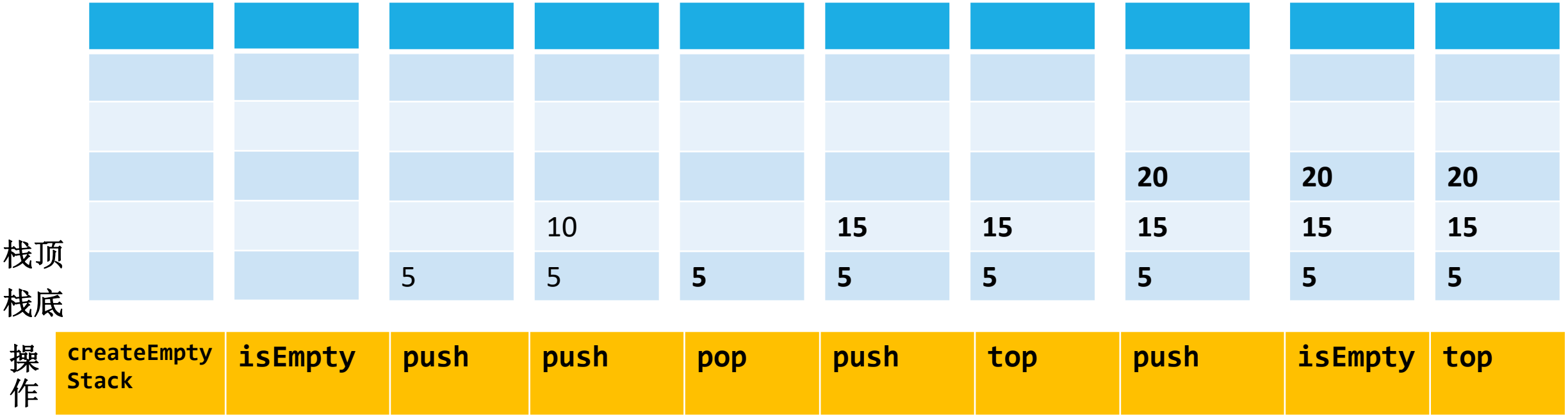
从队列que中删除一个元素

Datatype Front(Queue que)

求队头元素

End ADT Queue

栈举例


栈顶
栈底
10
5
5
15
15
5
15
15
5
20
15
5
20
15
5
操作
createEmpty Stack
isEmpty
push
push
pop
push
top
push
isEmpty
top

栈混洗

n个数据$(a_1, a_2, \ldots, a_n)$依次进栈，并随时可能出栈，按照其出栈次序得到的每一个序列$(a_{k1}, a_{k2}, \ldots a_{kn})$，称为一个栈混洗。

现在考虑有三个元素（i, j, k）按照先后次序压入栈中，则可能的栈混洗有：(i, j, k)、(k ,j, i)、(i, k, j)、(j, i, k)、(j, k, i)。(k, i, j)必然非栈混洗。

栈混洗	操作1	操作2	操作3	操作4	操作5	操作6
(i, j, k)	push(i)	pop(i)	push(j)	pop(j)	push(k)	pop(k)
(k ,j, i)	push(i)	push(j)	push(k)	pop(k)	pop(j)	pop(i)
(i, k, j)	push(i)	pop(i)	push(j)	push(k)	pop(k)	pop(j)
(j, i, k)	push(i)	push(j)	pop(j)	pop(i)	push(k)	pop(k)
(j, k, i)	push(i)	push(j)	pop(j)	push(k)	pop(k)	pop(i)