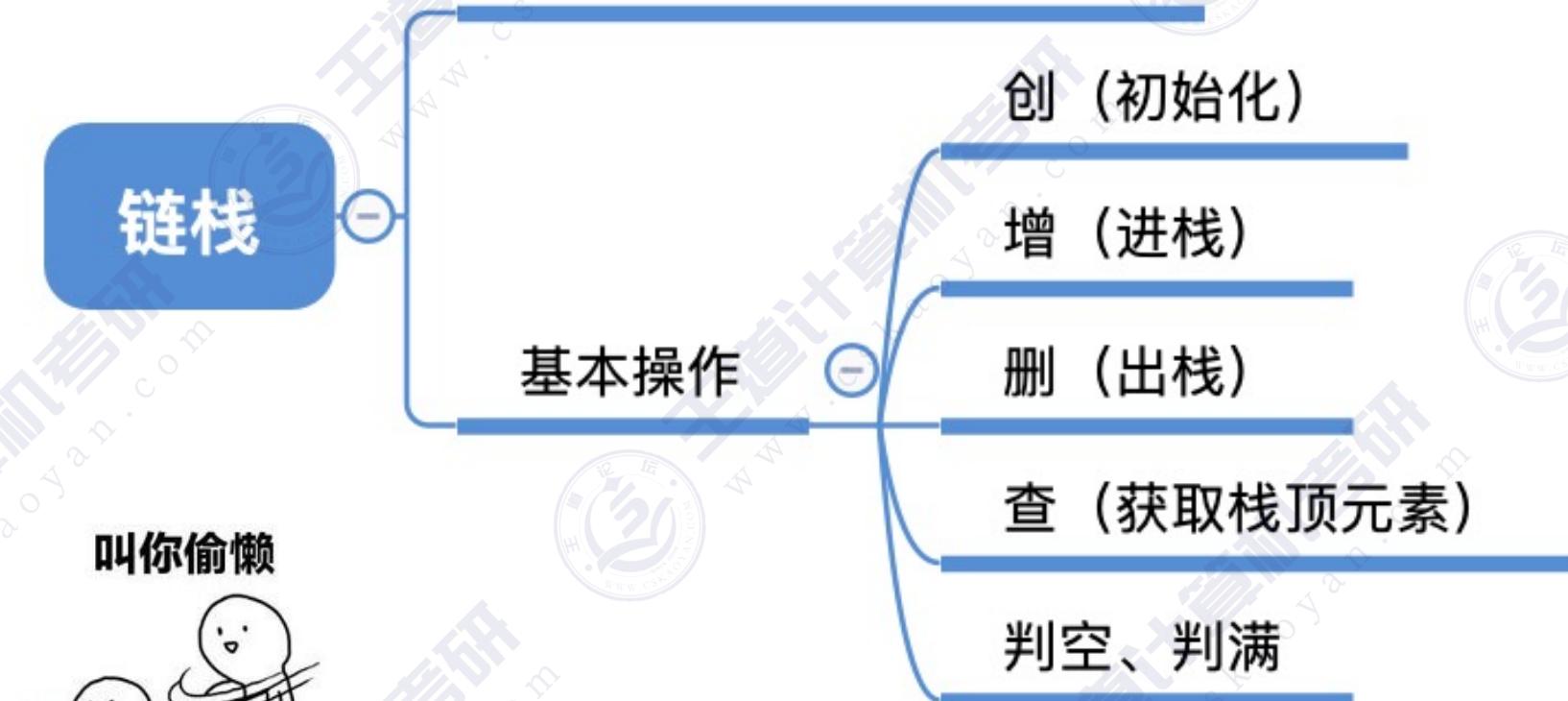


本节内容

链栈 的实现

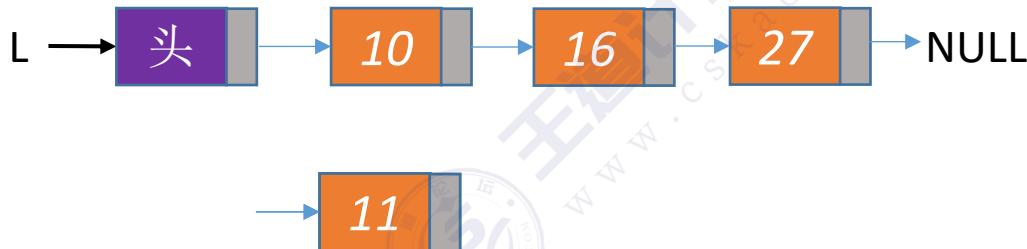
知识总览

用**链式存储**方式实现的栈



穿越：头插法建立单链表

对头结点的
后插操作



//后插操作：在p结点之后插入元素 e

```
bool InsertNextNode (LNode *p, ElemType e){  
    if (p==NULL)  
        return false;  
    LNode *s = (LNode *)malloc(sizeof(LNode));  
    if (s==NULL)    //内存分配失败  
        return false;  
    s->data = e;          //用结点s保存数据元素e  
    s->next=p->next;  
    p->next=s;          //将结点s连到p之后  
    return true;  
}
```

对应：进栈

叫你偷懒



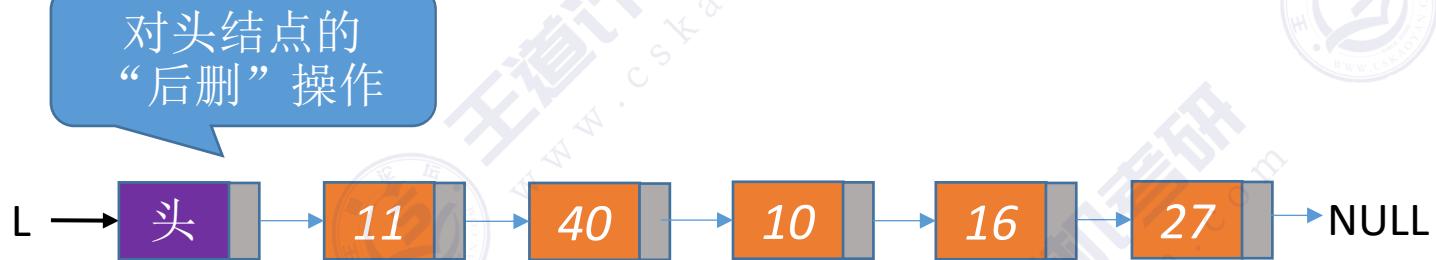
头插法建立单链表：

初始化单链表

```
While 循环 {  
    每次取一个数据元素 e;  
    InsertNextNode (L, e);  
}
```

穿越：单链表的删除操作

对应：出栈



链栈的定义

```
typedef struct Linknode{  
    ElemtType data;  
    struct Linknode *next;  
} *LiStack;
```

//数据域
//指针域
//栈类型定义

带头结点的初始化



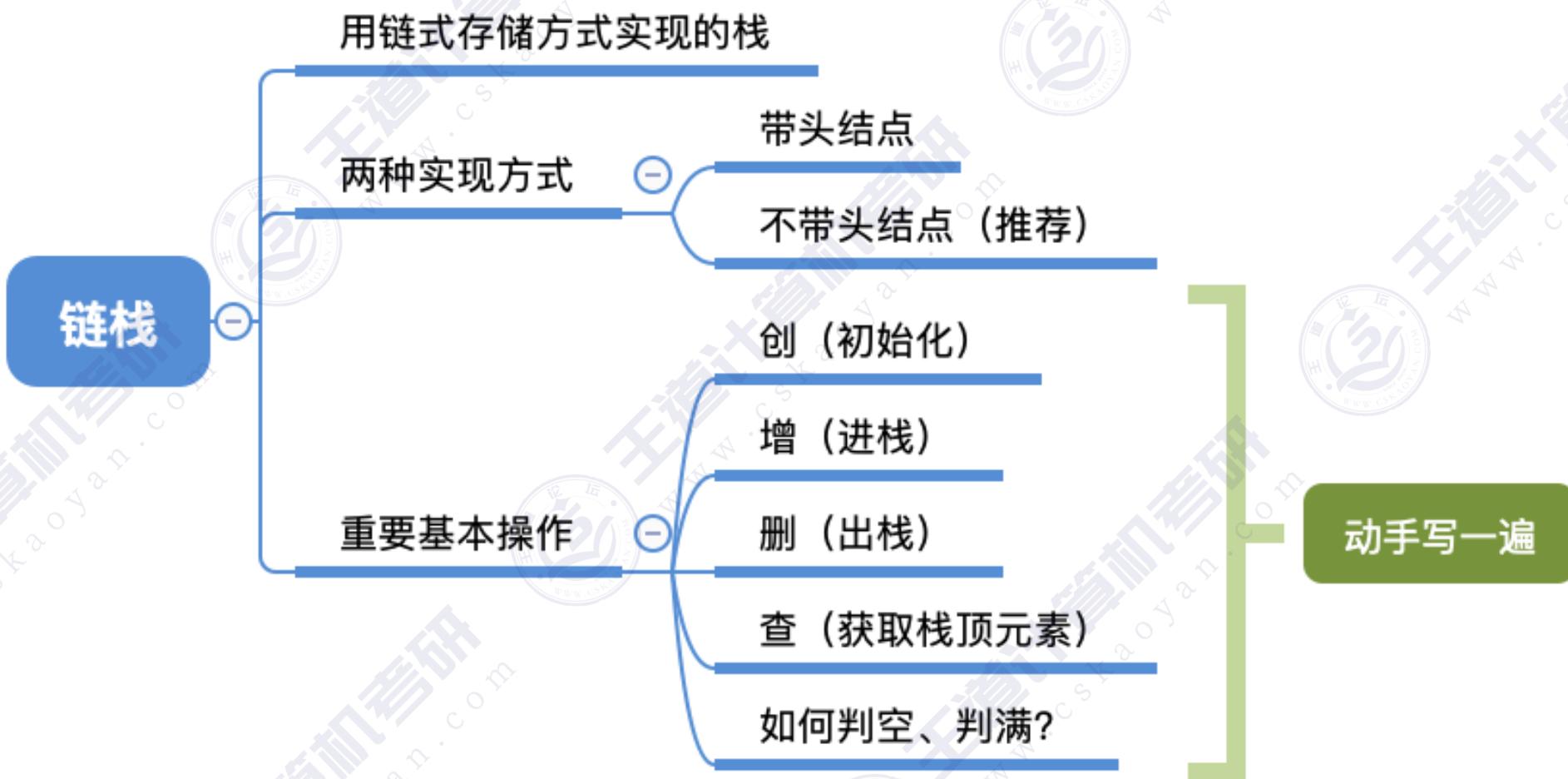
不带头结点的初始化



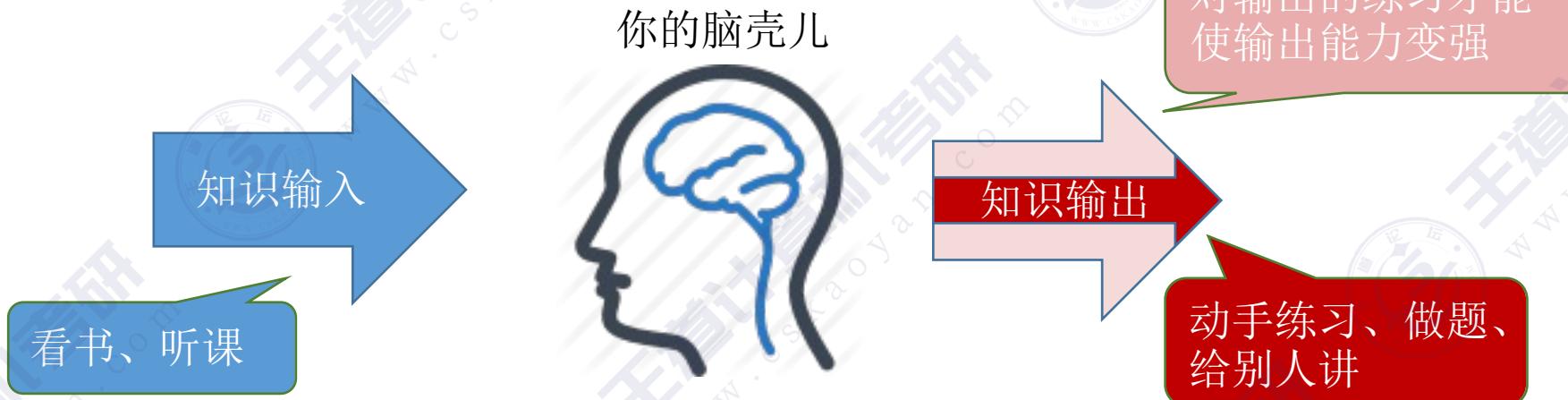
进栈/出栈都只能在栈顶一端进行
(链头作为栈顶)



知识回顾与重要考点



小建议



考试比的是“输出”的能力！



一根直肠通大脑
学到多少拉多少