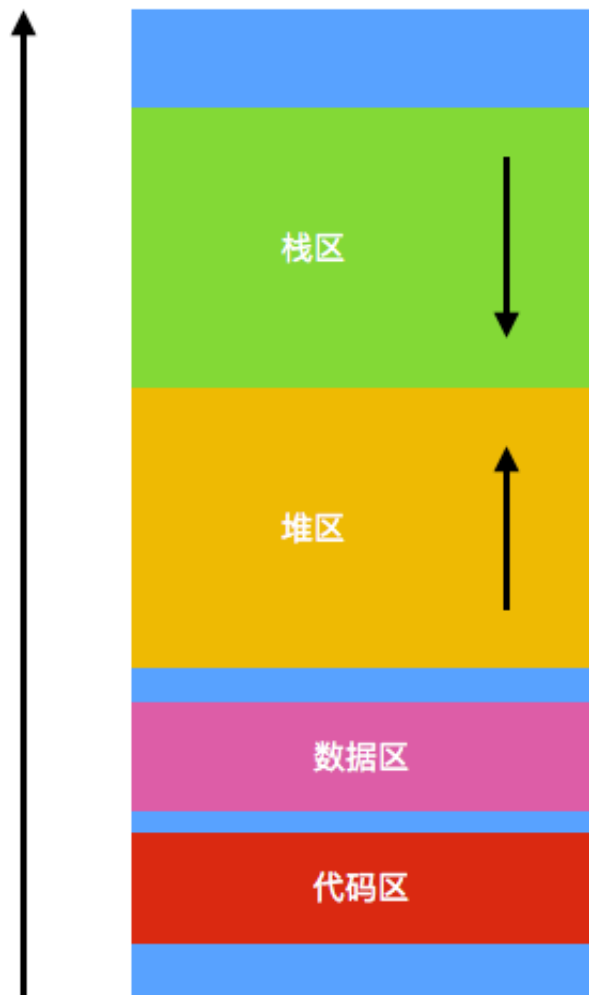


# 动态内存分配

---

## C程序的内存结构



- 栈区(stack): 主要用于函数调用过程中的内存分配(所有的普通局部变量, 所有的形式参数)
- 堆区(heap): 主要用于动态内存分配
- 数据区: 全局变量(已初始化部分和未初始化部分)、静态局部变量(static类型的局部变量)
- 文本区: 代码段, 存放二进制形式的指令(函数指令)

## 动态内存相关函数

```
#include <stdlib.h>

void *malloc(size_t size);
void free(void *ptr);
```

malloc函数用于动态申请指定字节大小的空间，申请成功，返回指向该空间的首地址；申请失败，返回NULL。

free函数用于释放malloc申请到的空间，其参数必须是malloc的返回值。

## 作业：查询并了解以下相关概念

- 了解log4c
- 全局变量：定义在函数之外的变量，全局变量被所有函数可见。
- static类型的全局变量：表示本文件私有
- static类型的局部变量：表示有当前函数的所有调用共享
- static的函数：表示该函数属于本文件私有
- const关键字：用于对内存空间进行制度操作限制
- const修饰的变量：修饰该变量为只读，即不可更改该变量的值
- const修饰的形式参数：修饰形式参数为只读，即不可更改该变量

```
const int num = 10;
const char *p;
char const *p;
char * const p;
const char * const p;
void function(const char *str);
```

今日目标：什么是顺序存储？什么是链式存储？什么是单链表？如何使用C语言实现？

## 链式存储----单链表

线性结构：逻辑上所有的数据除了第一个元素之外，每一个数据元素有且只有一个直接前驱，有且只有一个后继结点，第一个元素有且只有一个后继结点，最后一个元素有且只有一个前驱结点。

线性结构是一种非常简单的一对一的数据结构

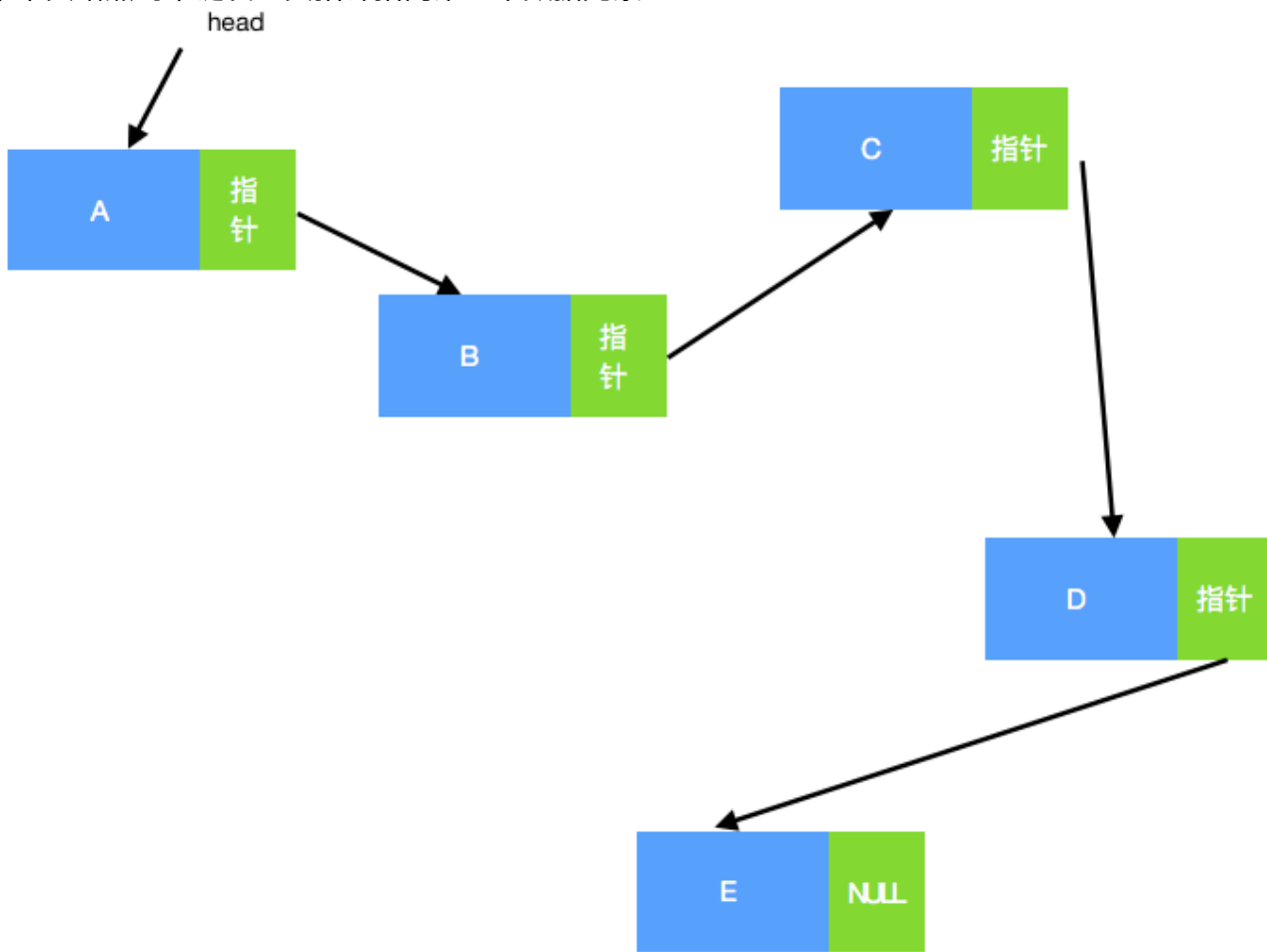
存储线性结构的时候，需要同时存储数据部分和关系部分

- 顺序存储：数据在逻辑上相邻在物理上也相邻，在C语言中可以使用数组来模拟顺序存储
- 链式存储：数据在逻辑上相邻，在物理上不必相邻，仅仅通过指向关系来确定逻辑上的前后关系。

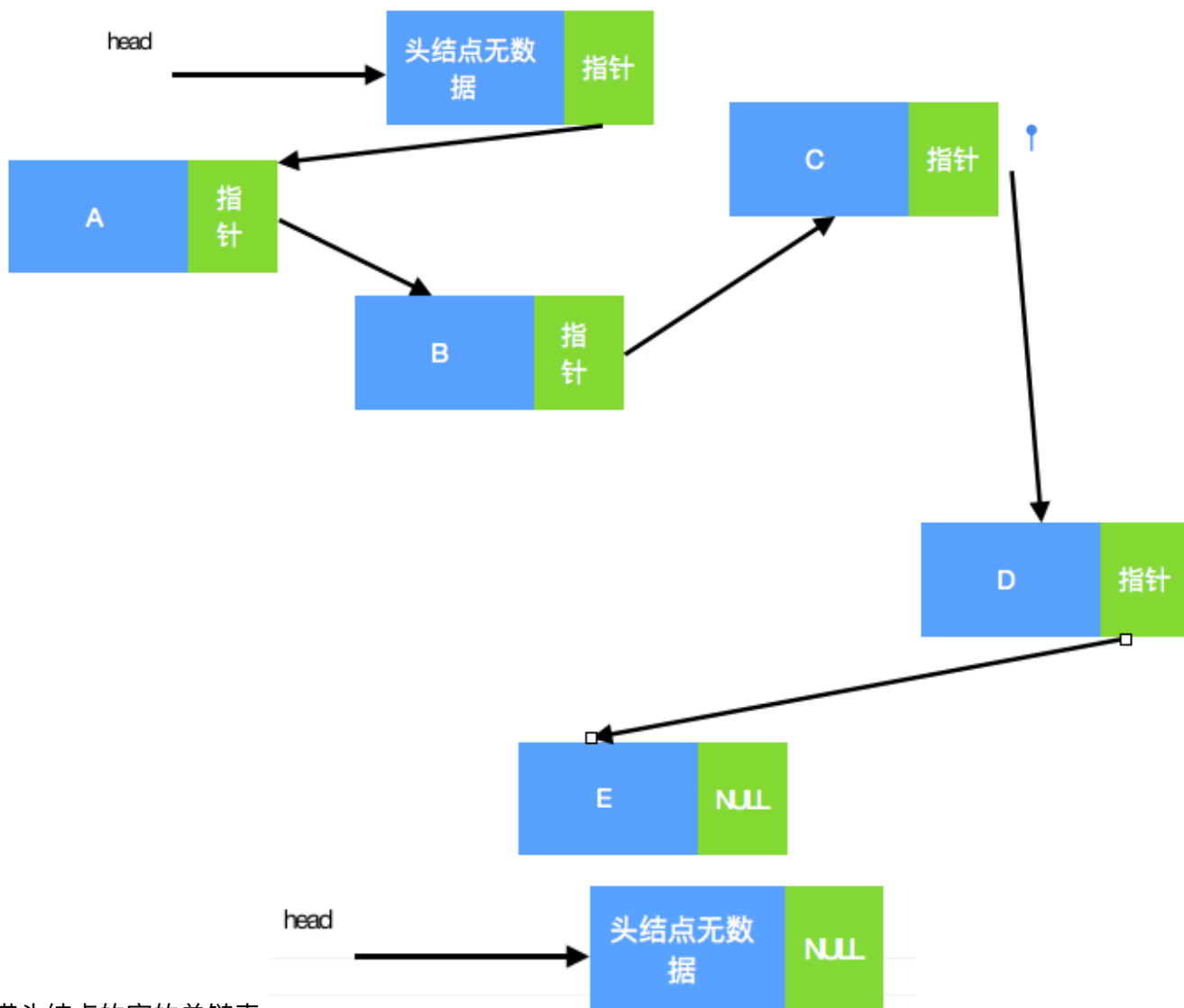
链式存储其实质是让数据分散在内存中的不同位置，通过指针完成数据之前前后后关系的引用，以保存灵活分配内存。

## 链式存储的实现

- 不带头结点的单链表：头指针指向第一个数据元素



- 带头结点的单链表：头指针指向头结点，头结点的作用是让插入和删除操作具备一致性（无论单链表是否为空，插入和删除操作均是在一个结点后进行）



- 带头结点的空的单链表

## 单链表的抽象数据类型

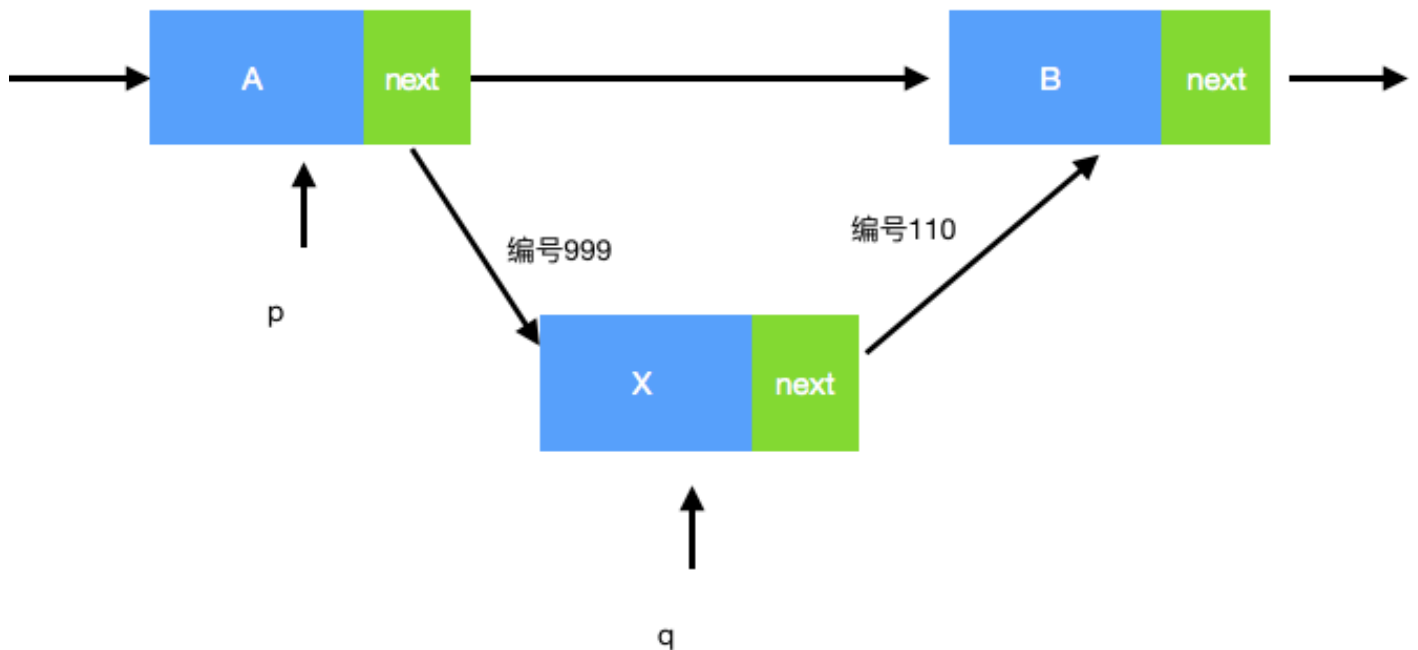
```

ADT link_list
{
    存储: 数据, 关系
    操作:
    1. 创建一个空的单链表
    2. 销毁单链表
    3. 求单链表的表长
    4. 判断单链表是否为空
    5. 遍历打印单链表
    6. 向单链表中插入元素 (*)
    7. 从单链表中删除元素 (*)
    8. 清空单链表
    9. 单链表逆序
    10. 头插
    11. 尾插
    12. 头删
    13. 尾删
    14. 获取单链表中指定位置的值
}

```

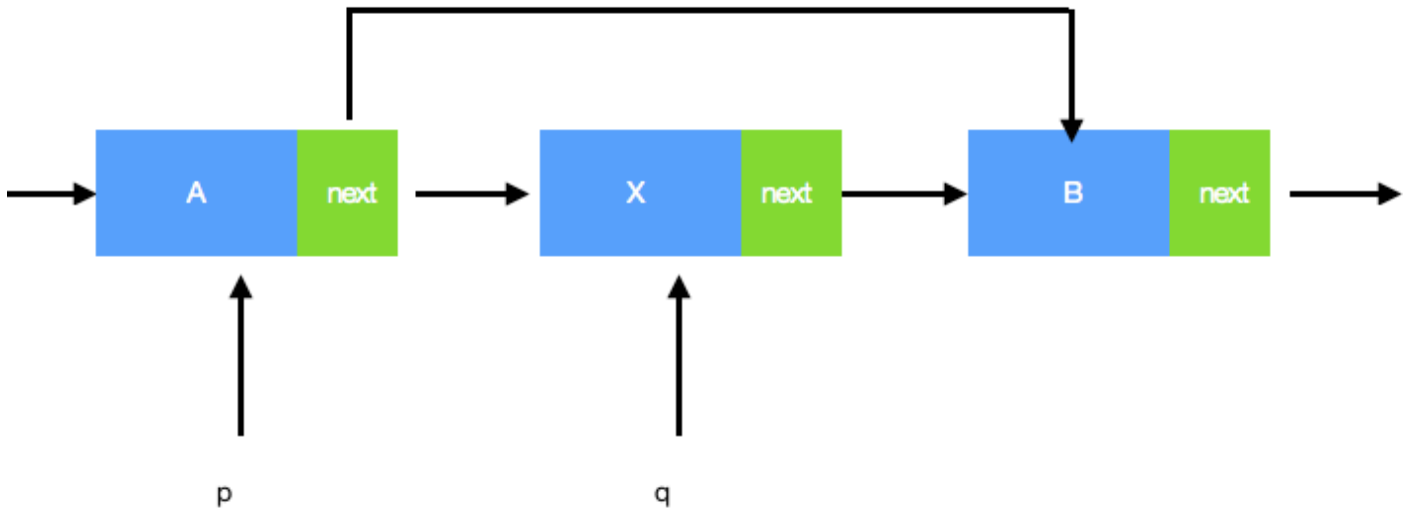
## 单链表的插入操作

编号110操作:  $q \rightarrow \text{next} = p \rightarrow \text{next};$   
 编号999操作:  $p \rightarrow \text{next} = q;$



## 单链表的删除操作

1. 记录待删除结点的位置:  $q = p \rightarrow \text{next};$
2. 将待删除结点从单链表中断开  
 $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next};$   
 $\Leftrightarrow$   
 $p \rightarrow \text{next} = q \rightarrow \text{next};$
3. 释放待删除的结点空间:  $\text{free}(q);$



## 单链表逆序

作业：完成上述所有的算法，并参考按位置插入，完成头插和尾插操作；参考按位置删除，完成头删和尾删

补充：

栈和队列都是操作受限的线性表

- 栈：规定只能在线性表的同一端进行插入和删除操作(栈顶)，另一端被称为栈底。栈结构的特点为"后进先出"。

对已完成的单链表代码只进行头插和头删，即可模拟实现一个链式栈。

- 队列：规定在线性表的一端进行插入操作(队尾)，在另一端进行删除操作(队首)。队列的特点为"先进先出"。

对已完成的单链表代码只进行头删和尾插，即可模拟实现一个链式队列。