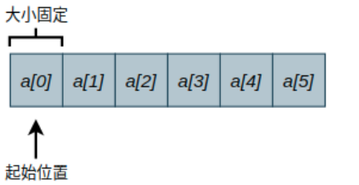
数组

定义

数组（Array）是一种线性表数据结构，使用一组连续的内存空间，来存储一组具有相同类型的数据。

- 线性表
  - 线性表，顾名思义，数组内元素元素像一根直线排布

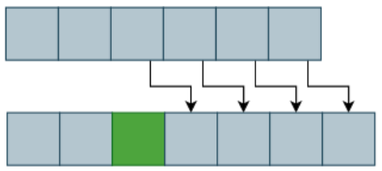
- 连续内存空间
  - 数组最为重要的特征，数组元素的排布是紧密相连的，中间不会产生空缺
    - 常说的内存连续指的是进程虚拟进程空间内存连续，而非物理内存空间连续
    - 操作系统采用MMU+页表来完成虚拟内存和物理内存的映射，只要虚拟内存空间连续，物理内存空间是否连续并不重要
- 相同类型数据
  - 相同数据类型决定了数组内的每一个元素所占用内存相等，为快速地数组下标访问提供了基础



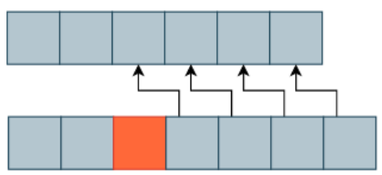
基于数组的三大特性，很容易得出左边的计算公式，这也是为什么数组元素使用下标访问时间复杂度只需要O(1)的原因: 只需要一次计算

元素的插入与删除

- 元素插入
  - 数组元素的插入就和打麻将时候摸牌插牌一样: 找到合适的位置，移动后面的牌产生空缺，将牌插入



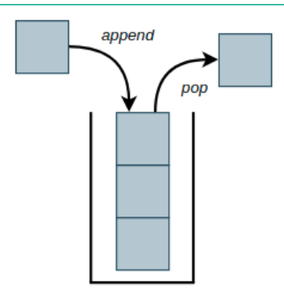
  - 为了保持数组中元素的原有顺序，在插入元素时，需要将后面的元素逐一向后挪一个位置，所以向数组中插入元素的效率并不高，平均时间复杂度为O(n)
  - 向数组末尾插入元素不需要移动任何元素，所以其时间复杂度为O(1)
- 元素删除
  - 数组元素的删除与元素插入类似，为了保证原有的元素顺序，需要移动数组内的元素



  - 与元素插入一样，数组内元素删除的平均时间复杂度为O(n)，但是删除末尾元素的时间复杂度为O(1)

数组的应用

- 实现容器
  - 在实际项目中，由数组作为核心部件所封装的列表(List)，使用更为频繁一些，最大的便利之处就在于支持动态扩容: 数组的大小一经确定，无法修改
- 实现栈
  - 由于在数据尾部添加和删除元素的时间复杂度均为O(1)，所以用数组来实现栈最好不过



- 实现字典
  - 利用数组O(1)的元素下标访问特性，可以利用哈希算法+数组来实现字典，只不过此类实现相较于树实现需占用更多内存