**1.顺序表存储（典型的数组）**  
     **原理：**顺序表存储是将数据元素放到一块连续的内存存储空间，相邻数据元素的存放地址也相邻（逻辑与物理统一）。  
     **优点：**（1）空间利用率高。（局部性原理，连续存放，命中率高）   
           （2）存取速度高效，通过下标来直接存储。  
    **缺点：**（1）插入和删除比较慢，比如：插入或者删除一个元素时，整个表需要遍历移动元素来重新排一次顺序。  
           （2）不可以增长长度，有空间限制,当需要存取的元素个数可能多于顺序表的元素个数时,会出现"溢出"问题.当元素个数远少于预先分配的空间时,空间浪费巨大。    
    **时间性能 :**查找 O(1) ,插入和删除O（n）。  
**2.链表存储**  
    **原理：**链表存储是在程序运行过程中动态的分配空间，只要存储器还有空间，就不会发生存储溢出问题，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点关系间的指针。  
   **优点：**（1）存取某个元素速度慢。   
          （2）插入和删除速度快，保留原有的物理顺序，比如：插入或者删除一个元素时，只需要改变指针指向即可。  
          （3）没有空间限制,存储元素的个数无上限,基本只与内存空间大小有关.   
    **缺点：**（1）占用额外的空间以存储指针(浪费空间，不连续存放，malloc开辟，空间碎片多)   
          （2）查找速度慢，因为查找时，需要循环链表访问，需要从开始节点一个一个节点去查找元素访问。  
    **时间性能 :**查找 O(n) ,插入和删除O（1）。   
**\*频繁的查找却很少的插入和删除操作可以用顺序表存储，堆排序,二分查找适宜用顺序表.  
\*如果频繁的插入和删除操作很少的查询就可以使用链表存储  
\*顺序表适宜于做查找这样的静态操作；链表适宜于做插入、删除这样的动态操作。  
\*若线性表长度变化不大，如果事先知道线性表的大致长度，比如一年12月，一周就是星期一至星期日共七天，且其主要操作是查找，则采用顺序表；若线性表长度变化较大或根本不知道多大时，且其主要操作是插入、删除，则采用链表，这样可以不需要考虑存储空间的大小问题。**

**\*顺序表:顺序存储，随机读取  
链式:随机存储,顺序读取(必须遍历)**

默认文件管理 nuatilus