1.在存取方式上：

顺序表可以顺序存放，也可以随机存放，链表只能从表头顺顺存取元素。如果说要存取第i个数据，顺序表只需要一次操作，而链表则需要i次操作。

1. 在逻辑机构与物理结构上

采用顺序存储时，逻辑上相邻的元素，对应的物理存储位置也相邻。反观采用链式存储的时候，逻辑上相邻的元素，存储时物理位置不一定相邻，用指针来表示对应的相邻关系。

1. 在空间分配上

顺序表在静态存储分配的情况下，一旦存满了就不能再扩充，如果再加入新元素，就会出现溢出，我称之为“数组爆了”，所以在用的时候，必须要给到足够大的内存空间，才能保证安全，但是同时可能又会导致大量的空间限制，造成浪费。顺序表在动态存储的时候，虽然说储存空间可以扩充，在扩充的同时需要移动大量的元素，会导致这个效率低下，如果内存中不存在更大块的连续的空间，也会导致分配失败。

反观链式存储的话，你需要的时候就申请，只要内存里面还有空间就可以申请，相比之下就更加高效、灵活。

1. 在实际情况下怎么选择两种方法
2. 基于存储方面的考虑：

如果说难以估计这个线性表的长度或者说你要去存储的数据的规模，这个时候就不太适合用顺序表去存储。链式存储的话，就不需要事先估计这个规模，只是说他的存储规模比较小。

1. 基于运算方面的考虑：

如果说需要经常做通过序号访问元素的话，很显然顺序表要优于链表。

当实现插入、删除的时候，在顺序表里面平均基本上得移动一半的数据，当数据量特别大的时候，这个运算量就不能忽视了，反观链表里面，你也需要找到位置，但是做的基本是比较操作，这样相比下来，显然后者要优于前者。

1. 基于环境方面的考虑：

并不是所有的语言都支持指针，比如Basic ，但是任何高级语言里面都有数组，顺序表实现起来就比较简单，反观链表的话，主要是操作指针，不支持指针的语言就只能考虑静态链表。

实现静态链表的时候，把本来的指针域的指针换成下一个数据的下标。