中缀表达式转换为后缀表达式算法：

➢ 建立运算符栈,并向栈底压入#作为结束符

➢ 从左向右依次读入表达式

➢ 如果是运算数, 则输出

➢ 如果是操作符则按下面操作

• 如果栈外运算符优先级高于栈顶元素优先级,栈外运算符入栈

• 如果栈外运算符优先级低于栈顶元素优先级,则栈顶运算符出栈输出

• 当栈外为), 栈内运算符退至(为止

• 当栈外为#, 栈内运算符退至#为止

后缀表达式计算算法：设操作数栈，从左往右读入，遇符号则出栈计算再入栈结果。

中缀表达式计算算法：

* + 设置操作数栈和运算符栈（压入结束符#）
  + 从左往右读入
  + 若运算符优先级高于栈顶运算符，入栈
  + 若运算符优先级低于栈顶运算符，两栈出栈运算

n>DE/(P+E)时，顺序表的空间效率高于链表。（其中n表示线性表中当前元素的数目，P表示指针的存储单元大小(通常为4字节)，E表示数据元素的存储单元大小，D表示可以在数组中存储的线性表元素的最大数目。）

对于双链表中的双指针开销，有一种节省空间的方法，尽管它会使实现变得复杂，而且速度稍微减慢：在一个指针域中存储两个指针值的异或，想得到其中一个必须先知道另一个。（这种思想另一个例子是无临时变量的两变量交换：a=a+b; b=a-b; a=a-b;）

对有相反的空间需求关系的两个栈，为节省空间，可充分利用顺序栈单向延伸的特性，使用一个数组来存储两个栈，每个栈从各自的端点向中间延伸。

有多个分支的算法时很难用迭代来代替递归，必须使用递归或者与递归等价的算法，如河内塔算法、遍历二叉树算法、归并排序和快速排序算法。

使用栈可以模拟递归。