Chapter 3 栈与队列

栈

栈：只允许在一端插入和删除的线性表，允许插入和删除的一端称为栈顶，另一端称为栈底

核心操作：取栈顶元素，出栈，入栈，时间复杂度均为 $O(1)$

顺序栈：数组实现，有最大长度

链式栈：链表实现，栈顶在链头，使用头插法，无栈满问题

与栈相关的算法

多进制输出（以 B 进制为例）：

- n % B 压入栈中
- n = n / B
- 重复上述两步直至 n=0
- 从栈中弹出数字

括号匹配：

- 从左向右扫描直至字符串结束
  - 遇到 '(' 则入栈
  - 遇到 ')' 则从栈中弹出一个 '('
- 如有弹空栈（右多）或结束时栈非空（左多）则括号不匹配

表达式计算：中缀转后缀，后缀计算

后缀计算：

- 顺序扫描每一项
  - 若为操作数，压栈
  - 若为操作符，从栈中弹出两个操作数，将计算结果压栈
- 结束后栈顶为计算结果

中缀转后缀：

- 顺序读入字符

    - 若为操作数，直接输出
    - 若为左括号，压栈
    - 若为操作符
        - 若其为左结合且优先级小于等于栈顶元素或其为右结合且优先级小于栈顶元素，将栈顶操作符出栈（循环直至栈空或不满足前述条件）
        - 将其入栈
    - 若为右括号，将操作符出栈直至遇到左括号，左括号出栈但不输出
- 若结束后栈非空，弹出剩余所有操作符

或，设置栈内优先数和栈外优先数

操作符	#	(	^	*, /, %	+, -	)
isp（栈内）	0	1	7	5	3	8
icp（栈外）	0	8	6	4	2	1

- 读入结束符 #
- 重复读入直至再次读入结束符 #
    - 若 ch 为操作数，直接输出
    - 若 ch 为操作符，栈顶操作符为 op
        - icp(ch) > isp(op)，ch 入栈，读入下一个 ch
        - icp(ch) < isp(op)，操作符退栈并输出
        - icp(ch) == isp(op)，退栈但不输出，若退出的为 '('，读入下一个 ch
- 结束后序列便为后缀表达式

表达式符号总数为 n，时间复杂度为 O(n)

走迷宫：使用栈来记录以及回溯，**路径存在性**

出栈序列种数：Catalan 数 $C_n = \frac{1}{n+1} \times \frac{2n!}{n! \times n!}$（即 $\frac{1}{n+1} \times C_{2n}^n$）

递归

递归：若一个对象部分地包含它自己，或用自己给自己定义，则称这个对象是递归的；若一个过程直接或间接地调用自己，则称这个过程是递归的调用过程

- 递归的定义：斐波那契数等
- 递归的数据结构：单链表

- 递归的问题解法：汉诺塔

递归函数：递归调用，回归求值，**必须有退出条件，化为非递归状况处理，且每次调用必须使参数的规模减小**，需保存局部变量，参数，地址。

递归调用并不高效：斐波那契的递归求法时间复杂度为 $O(2^n)$，而循环实现为 $O(n)$

单向递归和尾递归可转化为迭代

队列

队列：只允许在一端插入，另一端删除的线性表，允许删除的一端称为队头，允许删除的一端称为队尾

核心操作：入队，出队，时间复杂度均为 $O(1)$

顺序存储的队列：循环队列，两个指针 front 与 rear，front 指向第一个元素的前一个位置，rear 指向最后一个元素。队空时为 `rear == front`，队满时为 `(rear+1) % maxSize == front`（与队空条件相区分）

链式队列：队头在链头，队尾在链尾，无队满问题，有队空问题（`front == NULL`）

队列的应用

打印二项展开式的系数：将每项相加的结果再次入队，且每行的数据以 0 为界

宽度优先搜索的实现：将每个结点的子结点入队列，具体应用如**寻找最短路径**

优先级队列

先进先出，每次取出优先级最高的元素（一般实际实现使用堆）