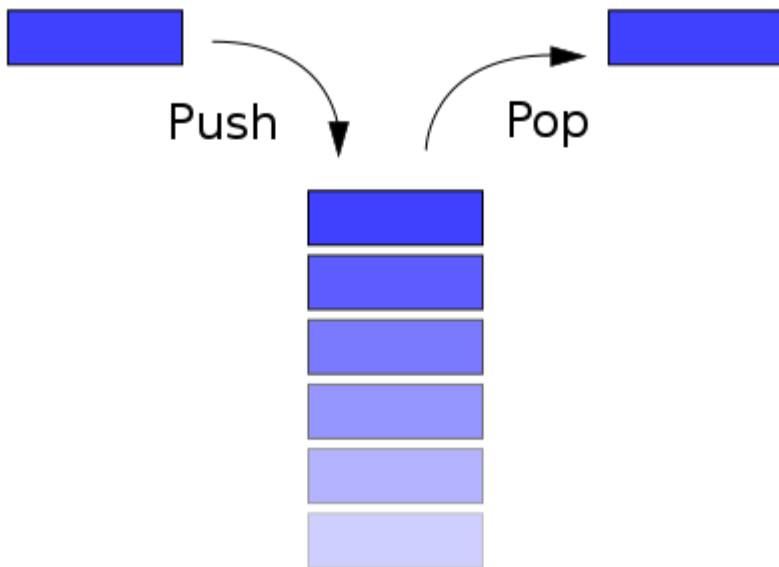


栈

栈（stack）是一种抽象数据类型，只允许在有序的线性数据集合的一端（称为堆栈顶端 top ）进行加入数据（push）和移除数据（pop）的运算。

栈按照后进先出（LIFO, Last In First Out）的原理运作。



操作

栈使用两种基本操作：推入（压栈，push）和弹出（弹栈，pop）：

- 推入：将数据放入栈顶端，栈顶端移到新放入的数据。
- 弹出：将栈顶端数据移除，栈顶端移到移除后的下一个数据。

特点

栈的基本特点：

1. 先入后出，后入先出。
2. 除头尾节点之外，每个元素有一个前驱，一个后继。

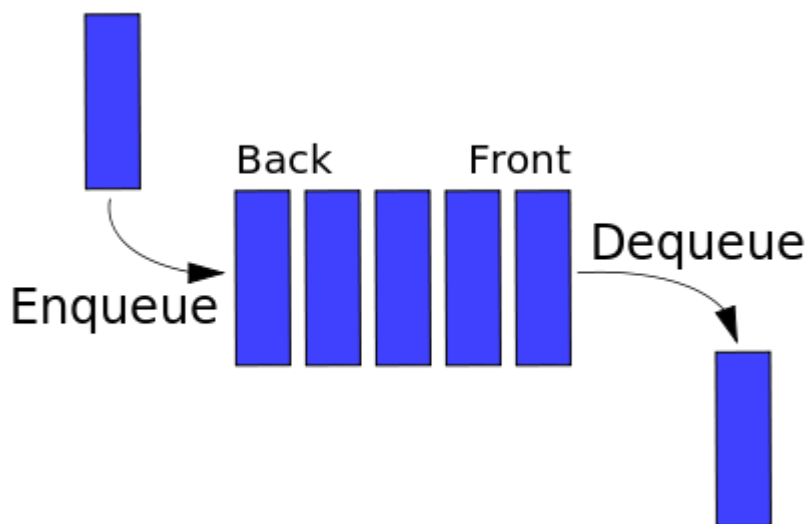
应用

- 回溯
- 递归
- DFS等

队列

队列（queue）是先进先出（FIFO, First-In-First-Out）的线性表。

队列只允许在后端（称为 $rear$ ）进行插入操作，在前端（称为 $front$ ）进行删除操作。



操作

队列使用两种基本操作：推入（入队，enqueue或push）和弹出（出队，dequeue或pop）：

- 推入：将数据放入队列尾部。
- 弹出：将队列顶端数据移除。

特点

队列的基本特点：

1. 先入先出，后入后出。
2. 除头尾节点之外，每个元素有一个前驱，一个后继。

逆波兰表达式

逆波兰表示法（Reverse Polish notation，RPN，或**逆波兰记法**）也被称为**后缀表示法**。

逆波兰记法中，操作符置于操作数的后面。例如表达“三加四”时，写作“3 4 +”，而不是“3 + 4”。如果有多个操作符，操作符置于第二个操作数的后面，所以常规中缀记法的“3 - 4 + 5”在逆波兰记法中写作“3 4 - 5 +”：先3减去4，再加上5。

中缀表达式转逆波兰表达式

1. 遇到操作数，直接输出；
2. 栈为空时，遇到运算符，入栈；
3. 遇到左括号，将其入栈；

- 4. 遇到右括号，执行出栈操作，并将出栈的元素输出，直到弹出栈的是左括号，左括号不输出；
- 5. 遇到其他运算符 +、-、*、/ 时，弹出所有优先级大于或等于该运算符的栈顶元素，然后将该运算符入栈；
- 6. 最终将栈中的元素依次出栈，输出。

经过上面的步骤，得到的输出既是转换得到的后缀表达式。

逆波兰表达式求值

例如：中缀表达式 $5 + ((1 + 2) * 4) - 3$ 写作逆波兰表达式为 `5 1 2 + 4 * + 3 -`

输入	操作	堆栈	注释
5	入栈	5	
1	入栈	5, 1	
2	入栈	5, 1, 2	
+	加法运算	5, 3	1, 2出栈，将结果3入栈
4	入栈	5, 3, 4	
*	乘法运算	5, 12	3, 4出栈，将结果12入栈
+	加法运算	17	5, 12出栈，将结果17入栈
3	入栈	17, 3	
-	减法运算	14	17, 3出栈，将结果14入栈