数据结构与算法

Data Structure and Algorithm

X. 堆栈和队列

授课人：Kevin Feng

翻译　：梁 少 华

Review

回顾

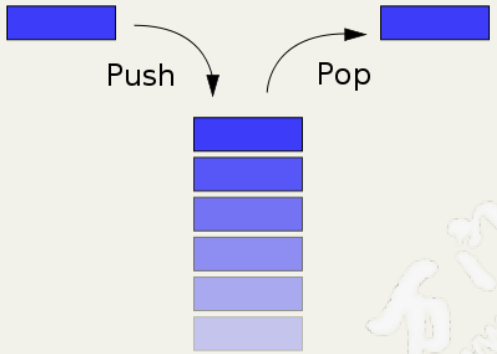
- 数据结构与算法
- 数学回顾
- 数组
- 数组列表
- 搜索和排列
- 递归与迭代
- 二进制搜索
- 分而治之
- 链接列表

概述

- 堆
  - 数据成员
  - 操作
- 队列
  - 数据成员
  - 操作

基本思想

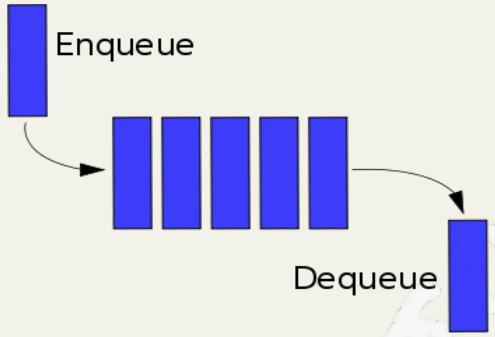
- 后进先出（LIFO）
- 操作:
  - 推:将项添加到堆栈的顶部,隐藏堆栈上已经存在的任何项
  - Pop:从堆栈顶部移除一个项目，并将此值返回给调用者



  - 堆栈通常用更多的操作来实现
  - 大小:返回堆栈中当前的项目数量
  - Peek:返回堆栈的当前顶层元素而不删除它
- 堆栈可以通过数组或链表轻松实现

队列基本思想

- 先进先出（FIFO）
- 操作:
  - Enqueue: 将一个项目添加到队列的末尾
  - Dequeue:从队列前面移除一个项目



  - 堆栈通常用更多的操作来实现
  - 大小:返回队列中当前的项目数量
  - Peek: 返回队列的当前顶层元素而不删除它。.
  - Peek: 返回堆栈的当前顶层元素而不删除它
- 队列可以通过（循环）数组或链表轻松实现

练习I

- 使用堆栈实现队列
- 使用队列实现堆栈
- 最小堆栈
  - 设计一个支持push,pop,top,并在常量时间内检索最小元素的堆栈.
  - getMin() -- 检索堆栈中的最小元素.
- 一个数组的两个堆栈
  - 描述如何使用单个数组来实现两个堆栈
- 一个数组的三个堆栈
  - 描述如何使用单个数组来实现三个堆栈

◉ 堆栈排序

- 编写一个程序,按升序排序堆栈.对于堆栈是如何实现的,你不应该做任何假设.下面是编写此程序时应该使用的唯一函数: push | pop | peek | isEmpty.

练习II

- 反转一个字符串
- 回文
- 有效的括号
  - 给定一个只包含字符 ‘(’, ‘)’, ‘{’, ‘}’, ‘[’ and ‘]’, 的字符串,确定输入字符串是否有效.
- 简化路径
  - 给定一个文件的绝对路径(Unix-style),简化它.
  - 例子,
    path = "/home/" => "/home"
    path = "/a/./b/../../c/" => "/c"

练习III

- 解码字符串
  - 给定一个编码字符串,返回它的解码字符串.
  - 编码规则是: k[encoded_string],其中方括号内的encoded_string正好重复了k次.请注意,k保证是一个正整数.
  - 你可以假定输入字符串总是有效的; 没有多余的空格,方括号格式正确等.
  - 此外，您可以假设原始数据不包含任何数字，并且这些数字仅用于那些重复数K。例如，不会有像3A或2(4)那样的输入.
  - 例子:
    - s = "3[a]2[bc]", return "aaabcbc".
    - s = "3[a2[c]]", return "accaccacc".
    - s = "2[abc]3[cd]ef", return "abcabccdcdcdef".

练习IV

- 棒球比赛
  - 你现在是棒球比赛记录员.
  - 给定一个字符串列表，每个字符串可以是以下4种类型之一:
    - 整数（一轮的得分）：直接表示您在本轮中获得的积分数.
    - “+”（一轮的得分）：表示您在本轮获得的积分是最后两轮有效积分的总和.
    - “D”（一轮的得分）：表示本轮获得的积分是最后一轮有效积分的两倍数据.
    - “C”（一种操作，不是一轮的得分）:表示您获得的最后一个有效回合的得分是无效的,应该被移除.
  - 每一轮的操作都是永久性的,可能会对前一轮和后一轮产生影响.
  - 你需要返回你在所有回合中得分的总和.
  - 输入: ["5","2","C","D","+"]
  - 输出: 30
    - 第1轮:你可以得到5分.总和是:5.
    - 第2轮:你可以得到2分. 总和为:7. 操作1:第2轮的数据无效. 总和是:5.
    - 第3轮:你可以得到10分（第二轮的数据已被删除）. 总数是:15。
    - 第4轮:你可以得到5 + 10 = 15分.
    - 总数是: 30.

练习V

- 行星碰撞
  - 我们给出了一组代表小行星的整数小星.
  - 对于每个小行星,绝对值代表它的大小,符号代表它的方向（正的意思是右边,负的意思是左边）.每个小行星都以相同的速度运动.
  - 找出所有碰撞后小行星的状.如果两个小行星相遇,较小的小行星就会爆炸.如果两者大小相同,两者都会爆炸.两个小行星在同一方向移动,永远不会相遇.
  - 例题 1:
    - 输入: asteroids = [5, 10, -5]
    - 输出: [5, 10]
    - 解释: 10和-5碰撞导致10次. 5和10不会相撞.
  - 例题2:
    - 输入: asteroids = [10, 2, -5]
    - 输出: [10]
    - 解释: 2和-5碰撞导致-5. 10和-5碰撞导致10.

练习 VI

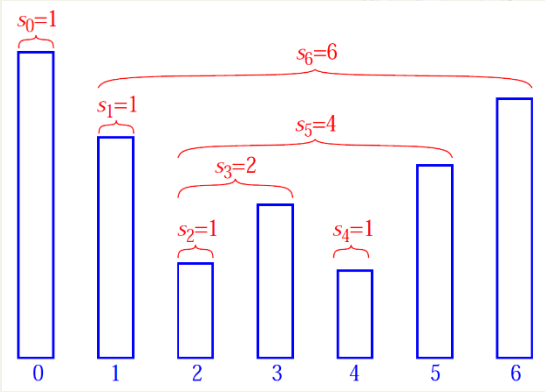
- 下一个大元素
  - 给定一个数组,为每个元素打印下一个更大的元素. 元素x的下一个较大元素x是数组x中右侧第一个较大元素。没有更大元素的元素,将下一个更大的元素视为-1.
- 下一个大元素 II
  - 给定一个循环数组（最后一个元素的下一个元素是数组的第一个元素）,为每个元素输出下一个更大的数字. 数字x的下一个较大数字是数组中下一个遍历顺序的第一个更大的数字,这意味着您可以循环搜索以找到下一个更大的数字. 如果不存在,则输出-1.

练习 VII

- 日常温度
  - 根据日常气温列表,制作一个列表,在输入的每一天中,都会告诉您需要等待多长时间,直到温度升高. 如果没有可能的将来的日子,请将0代替.
  - 例题,给出列表温度= [73, 74, 75, 71, 69, 72, 76, 73],你的输出应该是[1, 1, 4, 2, 1, 1, 0, 0].
- 双端队列 (双面堆栈/队列)
  - 堆栈允许只在一端插入和删除元素,而队列允许在一端插入并在另一端删除,而双端队列（双端队列）允许在两端插入和删除. 编写四个O(1)时间过,以将元素插入到从数组构造的双端队列的两端并从中删除元素.
- 滑动窗口最大值
  - 给定一个数组nums,有一个大小为k的滑动窗口,它从数组的最左边移动到最右边. 您只能在窗口中看到k个数字. 每次滑动窗口向右移动一个位置.
  - 例题,
    给定*nums* = [1, 3, -1, -3, 5, 3, 6, 7], 并且*k* = 3
  - 因此，返回最大滑动窗口为 [3, 3, 5, 5, 6, 7]

作业

- 最长有效子字符串的长度
  - 给定一个由开启和关闭括号组成的字符串,查找最长有效括号子字符串的长度.
- 股票跨度问题
  - 在特定的一天中,股票价格的跨度Si被定义为在给定日期之前连续几天的最大数量,当天股票的价格小于或等于 给定的一天.



项目

◉ 算术表达式评估

- 评估由一个字符串表示的表达式.表达式可以包含圆括号,你可以假定圆括号匹配良好. 为简单起见,你可以假设只允许二进制操作是+， - ， *和/.
- 算术表达式可以用三种形式之一来编写:
  - 中缀表示:操作符写在它们操作的操作数之间,例如,3 + 4 .
  - 前缀表示法:操作符在操作数之前写入.例如, + 3 4
  - 后缀表示法:运算符在操作数之后写入.例如, 3 4 +
- 从中缀转换为后缀
- 评估后缀表达式

回顾

- 栈和队列
  - 数组列表 实现
  - 链表 实现
- 样本面试问题

数据结构与算法

Data Structure and Algorithm

X. 堆栈和队列结束

授课人：Kevin Feng

翻译　：梁少华