

# 栈、队列、堆(优先队列)、贪心思想

ncst acm集训队  
林凯

2025 年 1 月 21 日

## Section 1

# 栈

# 栈的定义

栈是一种只允许在一端进行插入、删除的线性表。

栈底是固定的,不允许插入和删除的一端。栈顶是能够进行插入和删除的一端。所以栈是先进后出的线性表。

# 栈的操作

- StackInit:初始化栈
- StackPush:入栈
- StackPop:出栈
- StackTop:获取栈顶元素
- StackEmpty:检测栈是否为空
- StackDestroy:销毁栈

# 栈的实现(数组模拟)

- 用  $top$  表示栈顶所在的索引。初始时,

$$top = -1$$

- push  $x$ :

$$st[++top] = x$$

- pop:

$$top--$$

- empty:

$$top == -1 ? "YES" : "NO"$$

- query:

$$st[top]$$

# 栈的实现(STL)

- `empty()` 堆栈为空则返回真
- `pop()` 移除栈顶元素
- `push()` 在栈顶增加元素
- `size()` 返回栈中元素数目
- `top()` 返回栈顶元素

## Section 2

# 队列

# 队列的定义

队列是一种一端进行插入,另一端进行删除的线性表。

队头是进行删除操作的一端。队尾是进行插入操作的一端。所以队列是先进先出的线性表。



# 队列的操作

- QueueInit:初始化队列
- QueuePush:队尾入队列
- QueuePop:队头出队列
- QueueFront:获取队列头部元素
- QueueBack:获取队列队尾元素
- QueueEmpty:检测队列是否为空
- QueueDestroy:销毁队列

# 队列的实现(数组模拟)

- 用一个数组  $q$  保存数据
- 用  $hh$  代表队头,  $tt$  代表队尾
- pop:  $hh++$
- push: 先  $tt++$  后放入元素
- empty:  $tt \geq hh$  时非空
- query: 返回  $q[hh]$

# 队列的实现(STL)

- empty:检测队列是否为空
- size:返回队列中有效元素的个数
- front:返回队头元素
- back:返回队尾元素
- push\_back:在队列尾部入队列
- pop\_front:在队列头部出队列

## Section 3

### 堆(优先队列)

# 堆的定义

堆是一种树形结构,根是堆顶,堆顶是所有元素的最优值。堆有大根堆和小根堆,一般用二叉树实现。

# 堆的操作

- empty:返回堆是否为空
- top:返回根节点值
- push:插入元素
- pop:移除根节点并调整堆

## 堆的实现(数组模拟)

- heap\_swap:交换两个元素
- heap\_down:向下调整
- heap\_up:向上调整

## 堆的实现(STL - 优先队列)

- empty:检测堆是否为空
- size:返回堆中元素个数
- top:返回堆顶元素
- push:插入元素
- pop:删除堆顶元素



# 堆排序(Heap Sort)

- 1 构建最大堆
- 2 交换堆顶与最后一个元素
- 3 调整堆
- 4 重复上述步骤

## Section 5

# 贪心思想

# 贪心思想的定义

贪心算法指在每一步都采取最优的选择,不一定得到最优结果,但通常接近最优解。

# 贪心思想的性质

- 最优子结构
- 贪心选择性质

## 例题:Luogu P1223

**题意:** $n$ 个人排队接水,每个人接水的时间为  $t[i]$ ,求最短的平均等待时间。

## 例题:Luogu P1223

**思路:**选用接水时间短的人先接水。

**证明:** 假设有两个人a和b,a的接水时间小于b,如果a先接水,那么a的等待时间为0,b的等待时间为a的接水时间,如果b先接水,那么a的等待时间为b的接水时间,b的等待时间为0。所以应该选择等待时间最短的人先接水

## 例题:Luogu P1080

**题意:**国王邀请  $n$  位大臣玩游戏,需要重新安排顺序使最大奖赏尽可能少。

## 例题:Luogu P1080

**思路:**按每个大臣左右手乘积除以右手的值排序,然后按照排序后的顺序计算每个大臣的奖赏。

**证明:**假设有两个大臣a和b,a的左右手乘积除以右手的值小于b,如果a在b前面,那么a的奖赏为a的左右手乘积除以右手的值,b的奖赏为b的左右手乘积除以右手的值,如果b在a前面,那么a的奖赏为a的左右手乘积除以右手的值,b的奖赏为b的左右手乘积除以右手的值。所以应该按照左右手乘积除以右手的值排序



# The End