

剑指 Offer 09. 用两个栈实现队列

难度 简单

👍 272

☆ 收藏

🔗 分享

🌐 切换为英文

🔔 接收动态

📝 反馈

用两个栈实现一个队列。队列的声明如下，请实现它的两个函数 `appendTail` 和 `deleteHead`，分别完成在队列尾部插入整数和在队列头部删除整数的功能。（若队列中没有元素，`deleteHead` 操作返回 -1）

示例 1:

输入:

```
["CQueue","appendTail","deleteHead","deleteHead"]  
[[],[3],[],[[]]]
```

输出: [null,null,3,-1]

示例 2:

输入:

```
["CQueue","deleteHead","appendTail","appendTail","deleteHead","deleteHead"]  
[[],[],[5],[2],[],[[]]]
```

输出: [null,-1,null,null,5,2]

提示:

- `1 <= values <= 10000`
- 最多会对 `appendTail`、`deleteHead` 进行 `10000` 次调用

相关点

- 栈 队列的数据结构
- C++ 栈与队列底层实现

stack 先入后出

底层其实是利用顺序容器构造的一种容器适配器。

```
1 template<typename _Tp, typename _Sequence = deque<_Tp> >
```

很明显，实现栈是通过双端队列的接口来提供不同的功能接口。

- `top()`
- `push(valueType x)`
- `pop()`
- `swap(stack& _s)`
- `empty()`
- `emplace()`

内部点：栈是一种先入后出的结构，在C++ STL 库中，利用其已实现的`deque`构造这种栈。

也就是只在一端入，利用 `deque.push_back()`, `deque.pop_back()`, `deque.back()` `deque.empty()` `deque.emplace_back()`方法提供stack的方法。这其中根据C++版本不同，移动对象到栈顶，和构造对象等有所不同。

queue 队列 先入先出

queue 与 deque不同的是，这是一个单向的队列。

解决思路：

两个栈。

插入，只插入栈1。

删除 栈2 为空时，栈1不为空则把元素谈到栈2，栈2栈顶元素即为第一个入栈元素

栈2 为空，栈1为空，说明队列为空。

栈2 不为空，直接删除栈2内容。

C++ 代码实现：

头文件：

```
1 #include<iostream>
2 #include<stack>
3
4 template <typename T>
5 class Cqueue{
6 public:
7     Cqueue();
8     ~Cqueue();
9     void appendTail(const T& node);
10    T deleteHead();
11 private:
12    std::stack<T> s1;
13    std::stack<T> s2;
14 };
```

cpp文件：

```
1 template<typename T>
2 T Cqueue<T>::deleteHead()
3 {
4     //
```

```
5  if(s2.empty())
6  {
7  if(s1.empty())
8  {
9  return -1;
10 }
11 else{
12 while(!s1.empty())
13 {
14 s2.push(s1.top());
15 s1.pop();
16 }
17 }
18 }
19 tem = s2.top();
20 s2.pop();
21 return T;
22 }
23
```