

master ▾

...

leetcode-master / problems / 栈与队列理论基础.md



youngyangyang04 Update

History

1 contributor

97 lines (57 sloc) | 5.08 KB

PDF下载

代码随想录

刷题

微信群

B站

代码随想录

知识星球

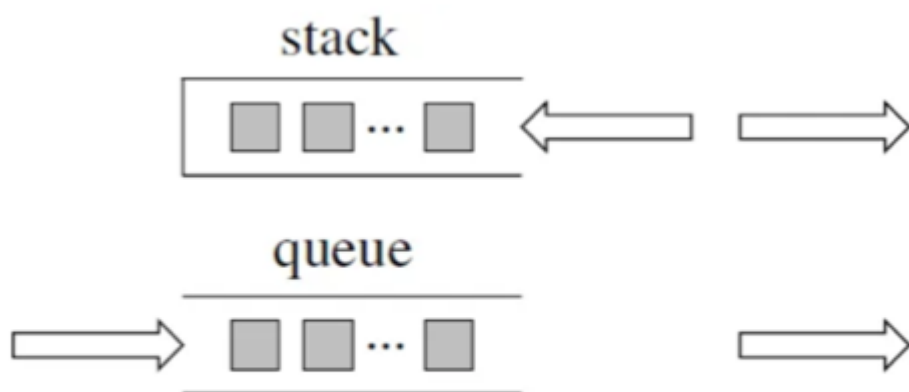
代码随想录

欢迎大家[参与本项目](#)，贡献其他语言版本的代码，拥抱开源，让更多学习算法的小伙伴们收益！

来看看栈和队列不为人知的一面

我想栈和队列的原理大家应该很熟悉了，**队列是先进先出，栈是先进后出**。

如图所示：



那么我这里在列出四个关于栈的问题，大家可以思考一下。以下是以C++为例，相信使用其他编程语言的同学也对应思考一下，自己使用的编程语言里栈和队列是什么样的。

1. C++中stack 是容器么？
2. 我们使用的stack是属于那个版本的STL？
3. 我们使用的STL中stack是如何实现的？
4. stack 提供迭代器来遍历stack空间么？

相信这四个问题并不那么好回答，**因为一些同学使用数据结构会停留在非常表面上的应用**，稍稍往深一问，就会有好像懂，好像也不懂的感觉。

**有的同学可能仅仅知道有栈和队列这么个数据结构，却不知道底层实现，也不清楚所使用栈和队列和STL是什么关系。**

所以这里我在给大家扫一遍基础知识，

首先大家要知道**栈和队列是STL（C++标准库）里面的两个数据结构。**

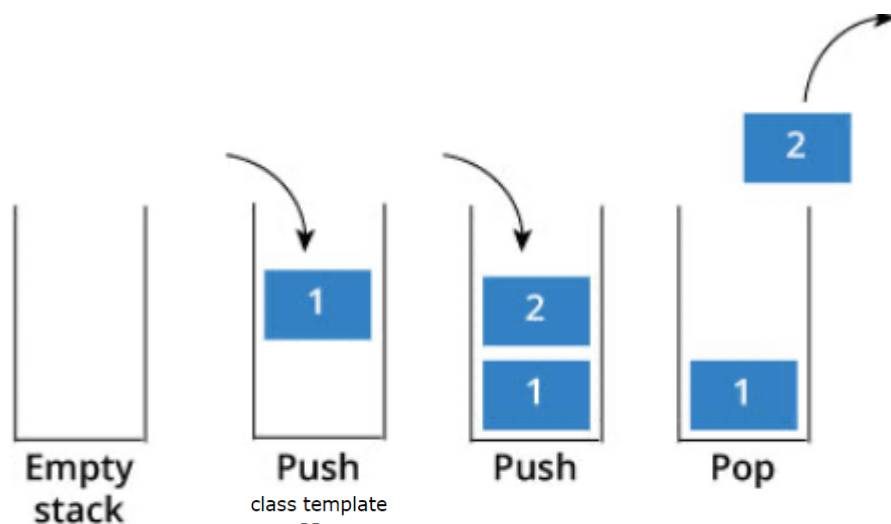
C++标准库是有多个版本的，要知道我们使用的STL是哪个版本，才能知道对应的栈和队列的实现原理。

那么来介绍一下，三个最为普遍的STL版本：

1. HP STL 其他版本的C++ STL，一般是以HP STL为蓝本实现出来的，**HP STL是C++ STL的第一个实现版本，而且开放源代码。**
2. P.J.Plauer STL 由P.J.Plauer参照HP STL实现出来的，**被Visual C++编译器所采用，不是开源的。**
- ✓ 3. **SGI STL** 由Silicon Graphics Computer Systems公司参照HP STL实现，**被Linux的C++编译器GCC所采用，SGI STL是开源软件，源码可读性甚高。**

接下来介绍的栈和队列也是SGI STL里面的数据结构，知道了使用版本，才知道对应的底层实现。

来说一说栈，栈先进后出，如图所示：



class template  
**std::list**

```
template < class T, class Alloc = allocator<T> > class list;
```

**List**

Lists are sequence containers that allow constant time insert and erase operations anywhere within the sequence, and iteration in both directions.

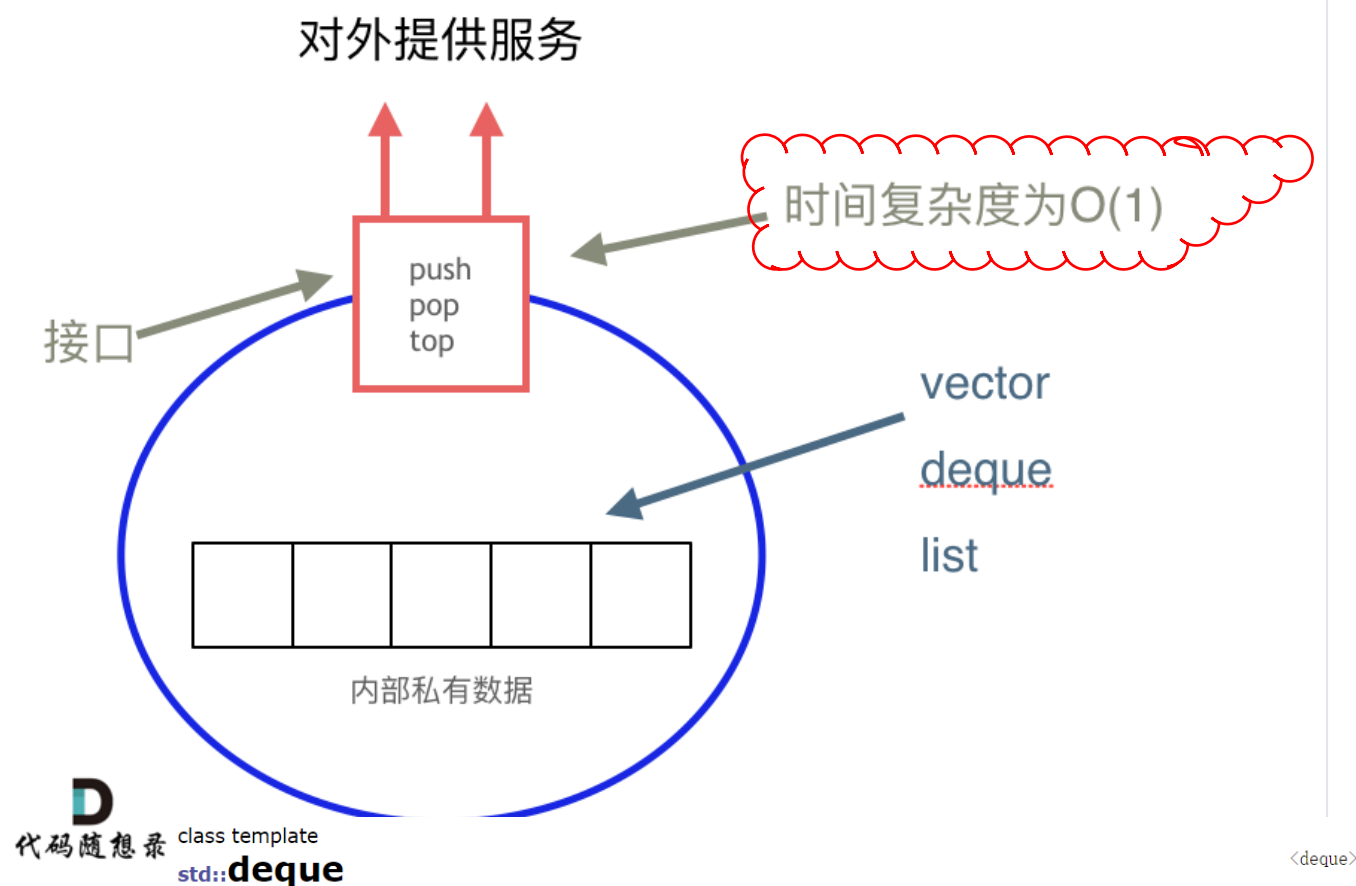
栈提供push 和 pop 等等接口，**所有元素必须符合先进后出规则，所以栈不提供走访功能，也不提供迭代器(iterator)。** 不像是set 或者map 提供迭代器iterator来遍历所有元素。 **遍历行为**

**栈是以底层容器完成其所有的工作，对外提供统一的接口，底层容器是可插拔的（也就是说我们可以控制使用哪种容器来实现栈的功能）。**

所以STL中栈往往不被归类为容器，而被归类为**container adapter（容器适配器）**。

那么问题来了，STL 中栈是用什么容器实现的？

从下图中可以看出，栈的内部结构，栈的底层实现可以是vector，deque，list 都是可以的，主要就是数组和链表的底层实现。



**deque** (usually pronounced like "deck") is an irregular acronym of **double-ended queue**. Double-ended queues are sequence containers with dynamic sizes that can be expanded or contracted on both ends (either its front or its back).

我们常用的SGI STL，如果没有指定底层实现的话，默认是以deque为缺省情况下栈的低层结构。

deque是一个双向队列，只要封住一段，只开通另一端就可以实现栈的逻辑了。

SGI STL中 队列底层实现缺省情况下一样使用deque实现的。

我们也可以指定vector为栈的底层实现，初始化语句如下：

```
std::stack<int, std::vector<int> > third; // 使用vector为底层容器的栈
```

刚刚讲过栈的特性，对应的队列的情况是一样的。

队列中先进先出的数据结构，同样不允许有遍历行为，不提供迭代器，SGI STL中队列一样是以deque为缺省情况下的底部结构。

也可以指定list 为起底层实现，初始化queue的语句如下：

```
std::queue<int, std::list<int>> third; // 定义以list为底层容器的队列
```

所以STL 队列也不被归类为容器，而被归类为container adapter（容器适配器）。

我这里讲的都是C++ 语言中情况，使用其他语言的同学也要思考栈与队列的底层实现问题，不要对数据结构的使用浅尝辄止，而要深挖起内部原理，才能夯实基础。

- 作者微信: [程序员Carl](#)
- B站视频: [代码随想录](#)
- 知识星球: [代码随想录](#)