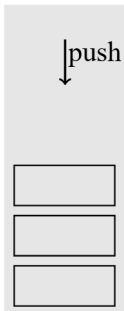


栈

引入



栈是 OI 中常用的一种线性数据结构。请注意，本文主要讲的是栈这种数据结构，而非程序运行时的系统栈/栈空间。

栈的修改与访问是按照后进先出的原则进行的，因此栈通常被称为是后进先出（last in first out）表，简称 LIFO 表。

Warning

LIFO 表达的是 **当前在容器** 内最后进来的最先出去。

我们考虑这样一个栈

```
1  push(1)
2  pop(1)
3  push(2)
4  pop(2)
```

如果从整体考虑，1 最先入栈，最先出栈，2 最后入栈，最后出栈，这样就成了一个先进先出表，显然是错误的。

所以，在考虑数据结构是 LIFO 还是 FIFO 的时候，应当考虑在当前容器内的情况。

使用数组模拟栈

我们可以方便的使用数组来模拟一个栈，如下：

实现

C++

```
1  int st[N];
2  // 这里使用 st[0] (即 *st) 代表栈中元素数量, 同时也是栈顶下标
3
4  // 压栈 :
5  st[++st] = var1;
6  // 取栈顶 :
7  int u = st[*st];
8  // 弹栈 : 注意越界问题, *st == 0 时不能继续弹出
9  if (*st) --*st;
10 // 清空栈
11 *st = 0;
```

Python

```
1  st = [0] * N
2  # 这里使用 st[0] 代表栈中元素数量, 同时也是栈顶下标
3
4  # 压栈 :
5  st[st[0] + 1] = var1
6  st[0] = st[0] + 1
7  # 取栈顶:
8  u = st[st[0]]
9  # 弹栈: 注意越界问题, *st == 0 时不能继续弹出
10 if st[0]:
11     st[0] = st[0] - 1
12 # 清空栈
13 st[0] = 0
```

C++ STL 中的栈

C++ 中的 STL 也提供了一个容器 `std::stack`, 使用前需要引入 `stack` 头文件。

STL 中对 `stack` 的定义

```
1 // clang-format off
2 template<
3     class T,
4     class Container = std::deque<T>
5 > class stack;
```

`T` 为 `stack` 中要存储的数据类型。

`Container` 为用于存储元素的底层容器类型。这个容器必须提供通常语义的下列函数：

- `back()`
- `push_back()`
- `pop_back()`

STL 容器 `std::vector`、`std::deque` 和 `std::list` 满足这些要求。如果不指定，则默认使用 `std::deque` 作为底层容器。

STL 中的 `stack` 容器提供了一众成员函数以供调用，其中较为常用的有：

- 元素访问
 - `st.top()` 返回栈顶
- 修改
 - `st.push()` 插入传入的参数到栈顶
 - `st.pop()` 弹出栈顶
- 容量
 - `st.empty()` 返回是否为空
 - `st.size()` 返回元素数量

此外，`std::stack` 还提供了一些运算符。较为常用的是使用赋值运算符 `=` 为 `stack` 赋值，示例：

```
1 // 新建两个栈 st1 和 st2
2 std::stack<int> st1, st2;
3
4 // 为 st1 装入 1
5 st1.push(1);
6
7 // 将 st1 赋值给 st2
8 st2 = st1;
9
10 // 输出 st2 的栈顶元素
11 cout << st2.top() << endl;
12 // 输出: 1
```

使用 Python 中的 list 模拟栈

在 Python 中，你可以使用列表来模拟一个栈：

实现

```
1  st = [5, 1, 4]
2
3  # 使用 append() 向栈顶添加元素
4  st.append(2)
5  st.append(3)
6  # >>> st
7  # [5, 1, 4, 2, 3]
8
9  # 使用 pop 取出栈顶元素
10 st.pop()
11 # >>> st
12 # [5, 1, 4, 2]
13
14 # 使用 clear 清空栈
15 st.clear()
```

参考资料

1. std::stack - zh.cppreference.com

 本页面最近更新：2023/10/15 13:01:10，[更新历史](#)

 发现错误？想一起完善？ [在 GitHub 上编辑此页！](#)

 本页面贡献者： [lr1d](#), [Xeonacid](#), [i-yyi](#), [william-song-shy](#), [iamtwz](#), [mgt](#), [Enter-tainer](#), [franklinqin0](#), [ksyx](#), [mcendu](#), [Menci](#), [nonPointer](#), [renbaoshuo](#), [shawlleyw](#), [Shen-Linwood](#)

© 本页面的全部内容 [在 CC BY-SA 4.0 和 SATA 协议之条款下](#) 提供，附加条款亦可能应用