

栈与队列

栈接口与实现

e4-A

邓俊辉

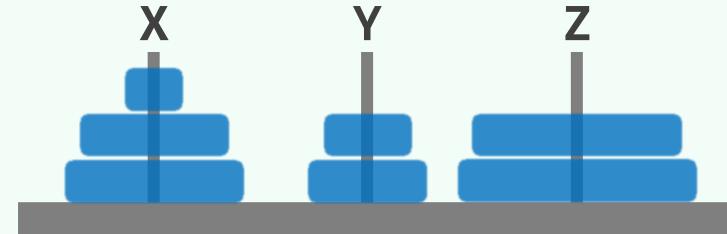
deng@tsinghua.edu.cn

陛下用群臣，如积薪耳，后来者居上

操作与接口

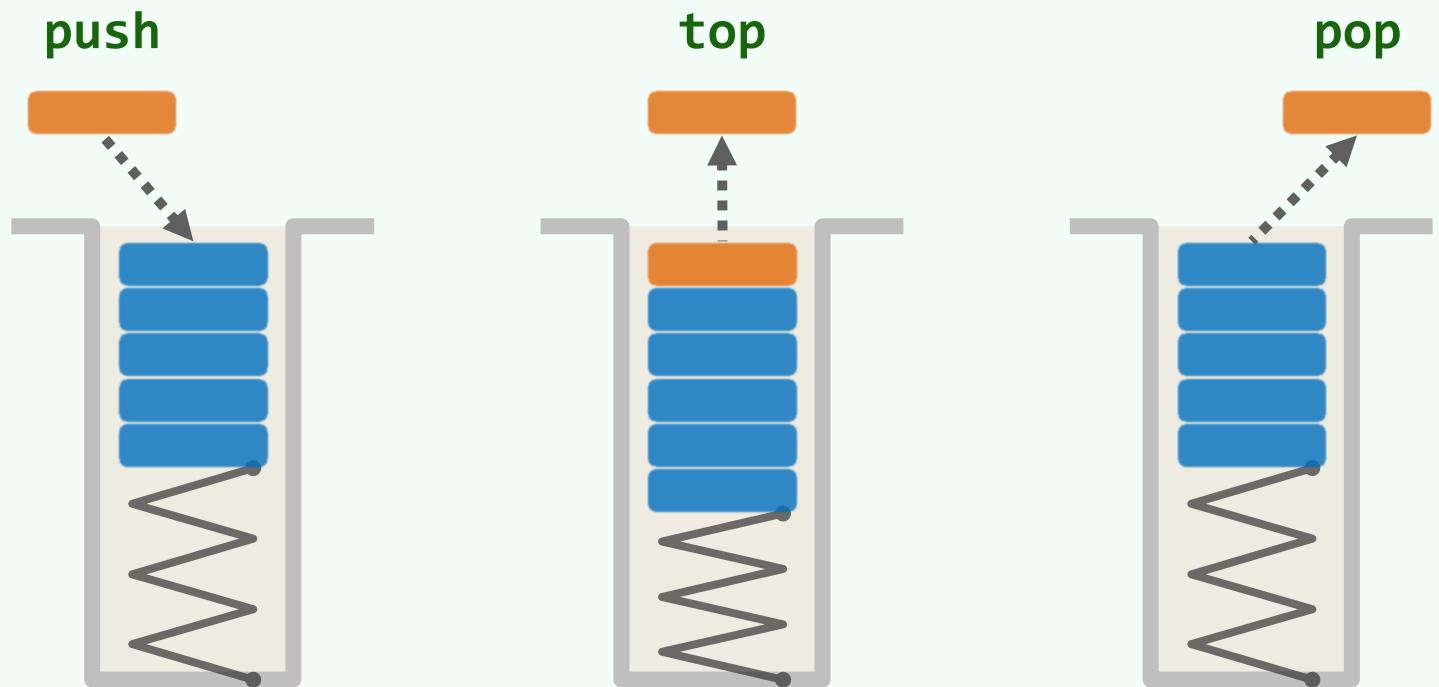
❖ 栈 (stack) 是受限的序列

- 只能在栈顶 (top) 插入和删除
- 栈底 (bottom) 为盲端



❖ 基本接口

- size() / empty()
- push() 入栈
- pop() 出栈
- top() 查顶



❖ 后进先出 (LIFO)

先进后出 (FILO)

❖ 扩展接口 : getMax() ...

实例

操作	输出	栈 (左侧栈顶)
Stack()		
empty()	true	
push(5)		
push(3)		
pop()	3	
push(7)		
push(3)		
top()	3	
empty()	false	

A diagram showing a stack structure. It consists of five vertical columns of boxes. The first four columns have two boxes each, and the fifth column has one box. The boxes are arranged such that the bottom box of each pair is to the left of the top box. The numbers 3, 7, and 5 are placed in the boxes. The top box contains the number 5. The bottom row of boxes contains the numbers 3, 7, 5, 3, and 7 from left to right.

操作	输出	栈 (左侧栈顶)
push(11)		
size()	4	
push(6)		
empty()	false	
push(7)		
pop()	7	
pop()	6	
top()	11	
size()	4	

A diagram showing a stack structure. It consists of four vertical columns of boxes. The first three columns have two boxes each, and the fourth column has one box. The boxes are arranged such that the bottom box of each pair is to the left of the top box. The numbers 11, 3, 7, and 5 are placed in the boxes. The top box contains the number 11. The bottom row of boxes contains the numbers 11, 3, 7, and 5 from left to right.

实现

◆ 栈既然属于序列的特例，故可直接基于向量或列表派生

◆ template <typename T> class Stack: public Vector<T> { //由向量派生的栈模板类

public: //size()、empty()以及其它开放接口均可直接沿用

void push(T const & e) { insert(e); } //入栈

T pop() { return remove(size() - 1); } //出栈

T & top() { return (*this)[size() - 1]; } //取顶

}; //以向量首/末端为栈底/顶——颠倒过来呢？

◆ 确认：如此实现的栈各接口，均只需 $\mathcal{O}(1)$ 时间

◆ 课后：基于列表，派生定义栈模板类；你所实现的栈接口，效率如何？