

Яндекс.Практикум, Алгоритмы: Спринт 1.

Базовые структуры данных

🕒 4 авг 2020, 01:01:56
старт: 1 авг 2020, 18:02:26

J. Стек - MaxEffective

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.08 секунд	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
GNU C11 7.3	0.015 секунд	64Mb		
руру4	0.095 секунд	64Mb		
GNU c++ 11 x32 4.9	0.015 секунд	64Mb		
Python 3.7.3	0.12 секунд	64Mb		
GNU c++ 4.9	0.015 секунд	64Mb		
gc go	0.11 секунд	64Mb		
Oracle Java 8	0.5 секунд	64Mb		
GNU c++17 7.3	0.018 секунд	64Mb		
Node JS 8.16	0.22 секунды	64Mb		

Реализуйте класс `StackMaxEffective`, поддерживающий операцию определения максимума среди элементов в стеке. Сложность операции должна быть $O(1)$. Для пустого стека операция должна возвращать `None`. При этом `push` и `pop` также должны выполняться за константное время.

Формат ввода

В первой строке записано одно число - количество команд, оно не превосходит 1000. Далее идут команды по одной в строке. Команды могут быть следующих видов:

`push x` - добавить число `x` в стек

`pop` - удалить число с вершины стека

`get_max` - напечатать максимальное число в стеке

Если стек пуст при вызове команды `get_max` нужно напечатать `None`, для команды `pop` - `error`.

Формат вывода

Для каждой команды `get_max` напечатайте результат её выполнения. Если стек пустой, для команды `get_max` напечатайте `None`. Если происходит удаление из пустого стека - напечатайте `error`.

Пример 1

Ввод

Вывод

Ввод	Вывод
10	error
pop	error
pop	4
push 4	None
push -5	
push 7	
pop	
pop	
get_max	
pop	
get_max	

Пример 2

Ввод	Вывод
10	None
get_max	error
push -6	None
pop	2
pop	
get_max	
push 2	
get_max	
pop	
push -2	
push -6	

Язык GNU c++17 7.3 ▼

Набрать здесь

Отправить файл

1

Отправить

Предыдущая

Следующая