Design a Stack With Increment Operation (/contest/weekly-contest-180/problems/design-a-stack-with-increment-operation/)

Submission Detail

34 / 34 test cases passed.

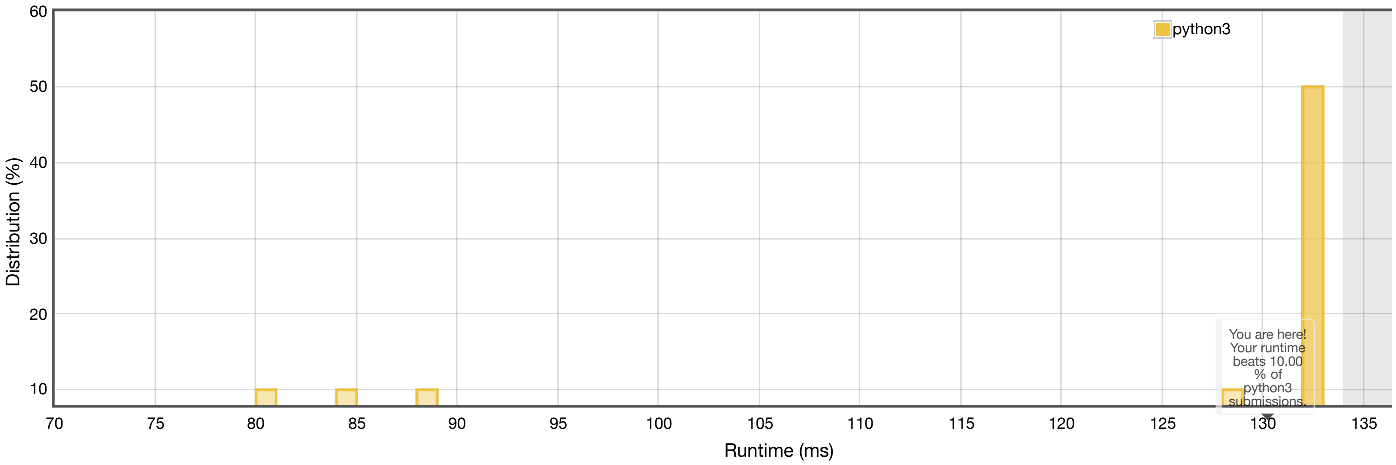
Status: Accepted

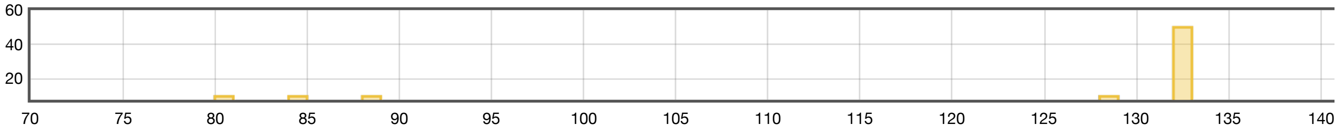
Runtime: **136 ms**

Memory Usage: **13.4 MB**

Submitted: **48 minutes ago**

Accepted Solutions Runtime Distribution





Zoom area by dragging across this chart

Accepted Solutions Memory Distribution

Sorry. We do not have enough accepted submissions to show distribution chart.

Invite friends to challenge **Design a Stack With Increment Operation**

2

Submitted Code: 48 minutes ago

Language: python3

Edit Code

```
class CustomStack:

    def __init__(self, maxSize: int):
        self.maxSize = maxSize
        self.elements = []

    def push(self, x: int) -> None:
        if len(self.elements) < self.maxSize:
            self.elements.append(x)

    def pop(self) -> int:
        if len(self.elements) == 0:
            return -1
        else:
            return self.elements.pop()    # Built in LOL
    def increment(self, k: int, val: int) -> None:
        if len(self.elements) < k:                  # Increment all numbers
            for i in range(len(self.elements)):
                self.elements[i] += val
        else:                                       # Increment first k nums
            for i in range(k):
                self.elements[i] += val


# Your CustomStack object will be instantiated and called as such:
# obj = CustomStack(maxSize)
# obj.push(x)
# param_2 = obj.pop()
# obj.increment(k,val)
```

Back to problem (/contest/weekly-contest-180/problems/design-a-stack-with-increment-operation/)

Help Center (/support/) | Jobs (/jobs/) | Bug Bounty (/bugbounty/) | Terms (/terms/) | Privacy Policy (/privacy/) (/region/)