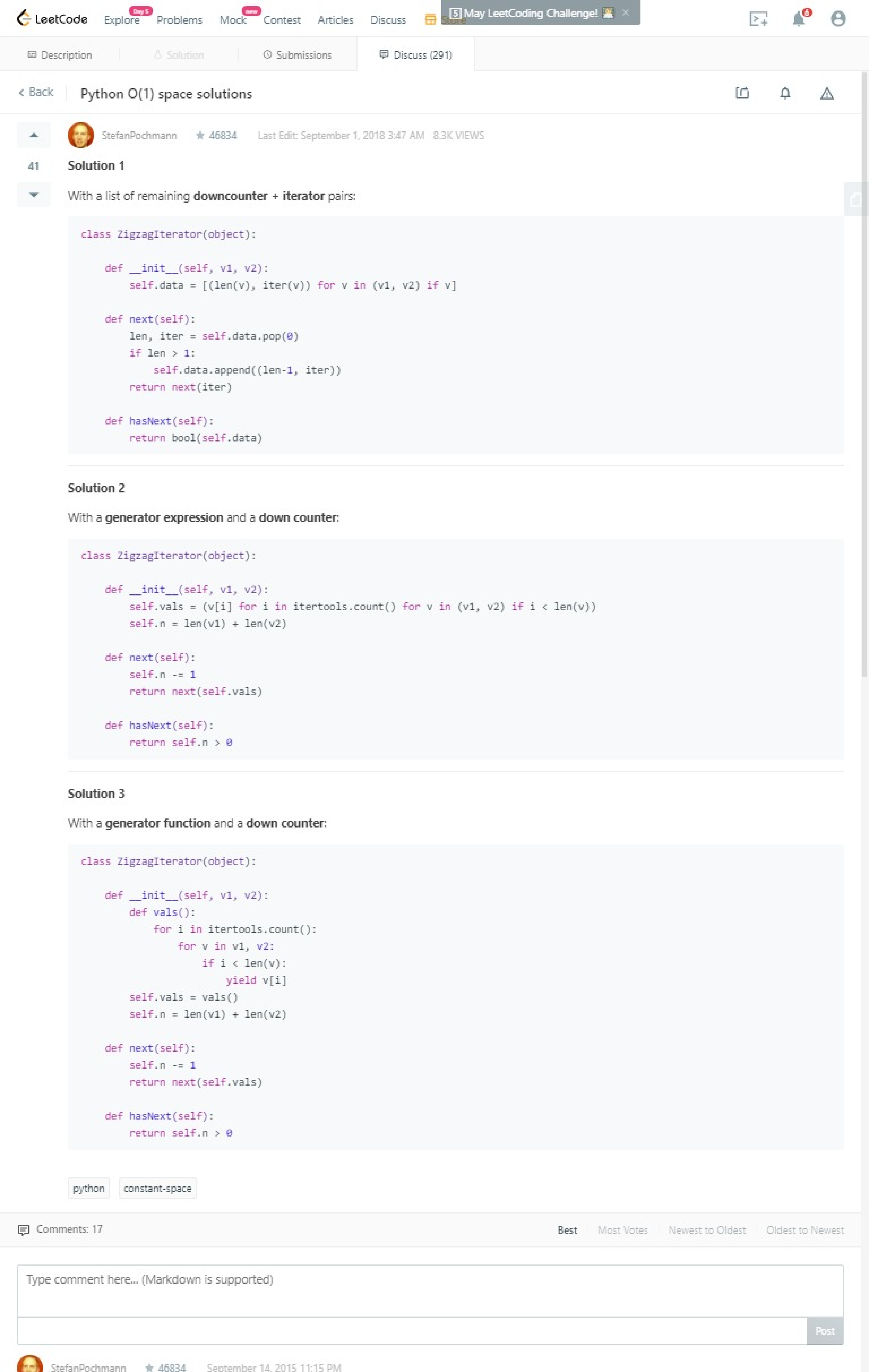

Python O(1) space solutions

StefanPochmann ★ 46834 Last Edit: September 1, 2018 3:47 AM 8.3K VIEWS

Solution 1

With a list of remaining **downcounter + iterator** pairs:

```
class ZigzagIterator(object):

    def __init__(self, v1, v2):
        self.data = [(len(v), iter(v)) for v in (v1, v2) if v]

    def next(self):
        len, iter = self.data.pop(0)
        if len > 1:
            self.data.append((len-1, iter))
        return next(iter)

    def hasNext(self):
        return bool(self.data)
```

Solution 2

With a **generator expression** and a **down counter**:

```
class ZigzagIterator(object):

    def __init__(self, v1, v2):
        self.vals = (v[i] for i in itertools.count() for v in (v1, v2) if i < len(v))
        self.n = len(v1) + len(v2)

    def next(self):
        self.n -= 1
        return next(self.vals)

    def hasNext(self):
        return self.n > 0
```

Solution 3

With a **generator function** and a **down counter**:

```
class ZigzagIterator(object):

    def __init__(self, v1, v2):
        def vals():
            for i in itertools.count():
                for v in v1, v2:
                    if i < len(v):
                        yield v[i]
        self.vals = vals()
        self.n = len(v1) + len(v2)

    def next(self):
        self.n -= 1
        return next(self.vals)

    def hasNext(self):
        return self.n > 0
```

StefanPochmann ★ 46834 September 14, 2015 11:15 PM