

Grader Output

ProblemSet3_S

3.01 / 3.01 points

earned

8 / 8 autograded

cells passed

Graded Cells

Cell 3 (cell-9bbed3588a21142)

Passed | 0.86 /

0.86 points

[View feedback](#)

Cell 6 (cell-b893840d1eabfeff)

Passed | 0.43 /

0.43 points

[View feedback](#)

Cell 6 (cell-450850f6c2af5c97)

Passed | 0 / 0

points

[View feedback](#)

Cell 6 (cell-8df6ac83ee3184c)

Passed | 0 / 0

points

[View feedback](#)

Assignment 3

Problem 1: Design a Correct Partition Algorithm

You are given code below for an incorrect partition algorithm that fails to partition arrays wrongly or cause out of bounds access in arrays. The comments include the invariants the algorithm wishes to maintain and will help you debug.

Your goal is to write test cases that demonstrate that the partitioning will fail in various ways.

```
In [1]: def swap(a, i, j):  
        assert 0 <= i  
        < len(a), f'accessing index {i} beyond end of array {len(a)}'  
        assert 0 <= j  
        < len(a), f'accessing index {j} beyond end of array {len(a)}'  
        a[i], a[j] = a[j], a[i]  
  
def tryPartition(a):
```

Cell 8 (cell-5e544036bd8af86

Passed | 0.86 / 0.86 points

[View feedback](#)

Cell 8 (cell-c3a88ce4fd6f93c8

Passed | 0 / 0 points

[View feedback](#)

Cell 8 (cell-7d56ccdf9924a6e

Passed | 0 / 0 points

[View feedback](#)

Cell 10 (cell-bd850b7eafce8a1

Passed | 0.86 / 0.86 points

[View feedback](#)

```
# implementat
ion of Lomuto par
titioning algorit
hm
```

```
n = len(a)
pivot = a[n-
1] # choose last
element as the pi
vot.
```

```
i, j = 0, 0 # i
nititalize i and j
both to be 0
```

```
for j in rang
e(n-1): # j = 0 t
o n-2 (inclusive)
    # Invaria
nt: a[0] .. a[i]
are <= pivot
```

```
#
a[i+1]...a[j-1] a
re > pivot
```

```
if a[j] <
= pivot:
```

```
    swap(
a, i+1, j)
    i = i
+ 1
```

```
    swap(a, i+1,
n-1) # place pivo
t in its correct
place.
```

```
return i+1 #
```