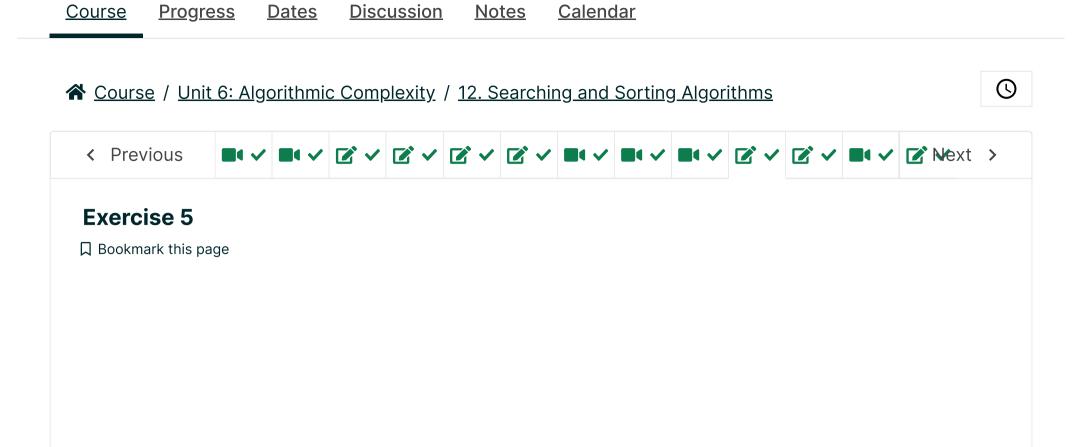
<u>Dates</u>

<u>Progress</u>

Discussion

<u>Help</u>

shengtatng ~



<u>Notes</u>

<u>Calendar</u>

Finger Exercises due Oct 27, 2022 07:30 +08 Completed

Exercise 5

3/3 points (graded)

ESTIMATED TIME TO COMPLETE: 8 minutes

Here is the code for selection sort. For simplicity, assume \Box is a list of integers:

```
def selSort(L):
 for i in range(len(L) - 1):
     minIndx = i
     minVal = L[i]
     j = i+1
     while j < len(L):
         if minVal > L[j]:
             minIndx = j
             minVal = L[j]
         j += 1
     if minIndx != i:
         temp = L[i]
         L[i] = L[minIndx]
         L[minIndx] = temp
```

And here is a suggested alternative:

```
def newSort(L):
 for i in range(len(L) - 1):
     j=i+1
     while j < len(L):
         if L[i] > L[j]:
             temp = L[i]
             L[i] = L[j]
             L[j] = temp
         j += 1
```

1. Do these two functions result in the same sorted lists?



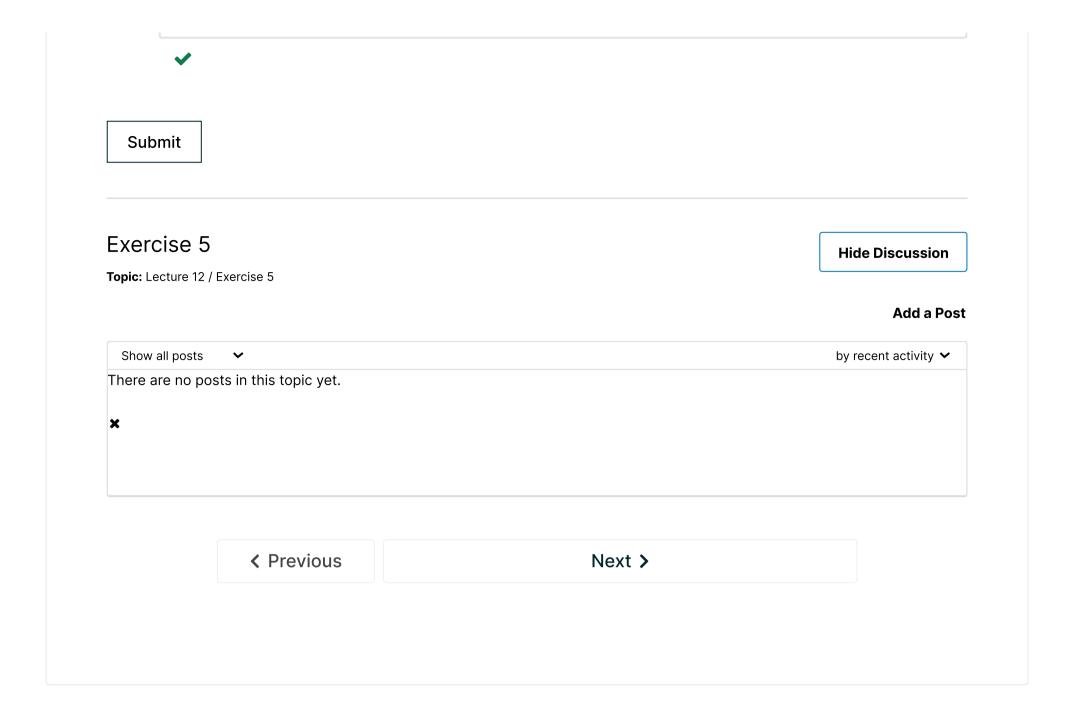
2. Do these two functions execute the same number of assignments of values into entries of the lists?	
	Yes. They execute the same number of assignments.
	No. newSort may use more - but never fewer - inserts than selSort.
	No. selSort may use more - but never fewer - inserts than newSort.
	No. Either function may use more inserts than the other.
	✓

3. Is the worst-case order of growth of these functions the same?

No. selSort has higher complexity than newSort.

```
Yes. newSort and selSort have the same complexity.
No. newSort has higher complexity than selSort.
```

Hide Notes



© All Rights Reserved



edX

About

<u>Affiliates</u>

edX for Business

Open edX

Careers

<u>News</u>

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

<u>Sitemap</u>

Connect

Blog

Contact Us

Help Center

Media Kit















© 2022 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>