

We are recovering from significant hosting issues. Much of the site is functional, but currently email delivery is not. Please bear with us as we validate site functionality.



Bookmarks

- ▶ [Welcome to the edX Platform](#)
- ▶ [Entrance Survey](#)
- ▶ [Download Python and Get Motivated!](#)
- ▶ [Week 1: Python Basics](#)
- ▶ [Week 2: Simple Programs](#)
- ▶ [Week 3: Structured Types](#)
- ▶ [Week 4: Good Programming Practices](#)
- ▶ [Midterm Exam](#)
- ▶ [Week 5: Object Oriented Programming](#)

Week 6: Algorithmic Complexity > Problem Set 6 > Problem 6

Problem 6

[Bookmark this page](#)

Problem 6-1

1 point possible (graded)

Answer the questions below based on the following sorting function. If it helps, you may paste the code in your programming environment. Study the output to make sure you understand the way it sorts.

```
def swapSort(L):  
    """ L is a list on integers """  
    print("Original L: ", L)  
    for i in range(len(L)):  
        for j in range(i+1, len(L)):  
            if L[j] < L[i]:  
                # the next line is a short  
                # form for swap L[i] and L[j]  
                L[j], L[i] = L[i], L[j]  
            print(L)  
    print("Final L: ", L)
```

Does this function sort the list in increasing or decreasing order? (items at lower indices being smaller means it sorts in increasing order, and vice versa)

☐ Increasing☐ Decreasing

You have used 0 of 1 attempt




▼ **Week 6:**
Algorithmic Complexity


11. Computational Complexity

[Finger Exercises](#) 

12. Searching and Sorting Algorithms

[Finger Exercises](#) 

Problem Set 6

[Problem Set due Mar 9, 2017 15:30 PST](#) 

► **Week 7:**
Plotting

► **Exit Survey**

► **Sandbox**

Problem 6-2

1 point possible (graded)

What is the worst case time complexity of `swapSort`? Consider different kinds of lists when the length of the list is large.

☐ $O(n^2)$

☐ $O(n)$

☐ $O(\log(n))$

☐ $O(1)$

Submit

You have used 0 of 1 attempt

Problem 6-3

1 point possible (graded)

If we make a small change to the line `for j in range(i+1, len(L)):` such that the code becomes:

```
def modSwapSort(L):
    """ L is a list on integers """
    print("Original L: ", L)
    for i in range(len(L)):
        for j in range(len(L)):
            if L[j] < L[i]:
                # the next line is a short
                # form for swap L[i] and L[j]
                L[j], L[i] = L[i], L[j]
            print(L)
    print("Final L: ", L)
```

What happens to the behavior of `swapSort` with this new code?

☐ No change



☐ `modSwapSort` now orders the list in descending order for all lists.

☐ `modSwapSort` now orders the list in descending order for SOME lists but not all

☒ `modSwapSort` enters an infinite loop.

Submit

You have used 0 of 1 attempt

Problem 6-4

1 point possible (graded)

What happens to the time complexity of this `modSwapSort` ?

☐ Best and worst cases stay the same.

☐ Worst case stays the same but best case changes.

☐ Best and worst cases change.

Submit

You have used 0 of 1 attempt

Problem 6

Topic: Problem Set 6 / Problem 6

Show Discussion

© All Rights Reserved





© 2012-2017 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX®

