**MITx: 6.00.1x Introduction to Computer Science and Programming Using ...**

## PROBLEM 6-1 (1/1 point)

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 1 of 1 submissions*

## PROBLEM 6-2 (1/1 point)

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

● $O(n^2)$ ✓

○ $O(n)$

○ $O(\log(n))$

○ $O(1)$

*You have used 1 of 1 submissions*

## PROBLEM 6-3 (1/1 point)

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.

*You have used 1 of 1 submissions*

## PROBLEM 6-4 (1/1 point)

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.

*You have used 1 of 1 submissions*

Show related resources ▼

About Blog News FAQs Contact Jobs Donate Sitemap