22C:16 Quiz 11

The two problems in this quiz involve writing a little bit of code - at most 4-6 lines each. If you see yourself writing too much, it is time to stop and think. Turn the page for the second problem.

1. You are given a list `L` of numbers and your task is to write a *recursive* function to determine if `L` is sorted in ascending order. Use the following function header:

   <div align="center">

   `def isSorted(L):`

   </div>

   and note that the function should return a boolean value, depending on whether `L` is sorted. For example, if `L` is `[3, 7, 7, 19, 21, 21]` then the function should return `True`.

   Of course this problem can be solved non-recursively, but you will not receive any credit for a non-recursive solution, even if it is correct. And, by the way, do not forget to specify the base cases.

   Hint: `L` is sorted if (i) the first item in `L` is less than or equal to the second item and (ii) the sublist of `L` excluding the first element is sorted.

2.

3. You are given a sorted list `L` of numbers and your task is to write a *recursive* function to determine two numbers in the list that are closest. Use the following function header:

<div align="center">

`def closest(L):`

</div>

For example, if `L` is `[3, 7, 67, 68, 210, 215]` then the function should return `[67, 68]`. Note that the object that is returned is a size-2 list.

Of course this problem can be solved non-recursively, but you will not receive any credit for a non-recursive solution, even if it is correct. And, by the way, do not forget to specify the base cases.

Hint: To find the closest pair of numbers in `L` first find the closest pair in the sublist of `L` that excludes the first element. Then you just have to compare this with the pair consisting of the first and second elements in `L`.