CSCA48 Exercise 7

Due: July 7, 2017. 5:00pm

This week we're going to start writing some recursive code. Don't worry, we'll start off slowly for now. Remember, plan your algorithm before you start writing. Writing code without a plan is a sure recipe for disaster when it comes to recursion.

Simple Recursion

In a file called `ex7.py` you must complete the following functions:

- `rsum`: Return the sum of all elements in a given list
- `rmax`: Return the maximum number in a given list
- `second_smallest`: Return the second smallest number in a given list [1]
- `sum_max_min`: Return the sum of the maximum and minimum elements in a given list

All functions will take a list of integers as input, all lists will have at least one element (2 in the case of `second_smallest`).

These functions seem pretty trivial, and they are. But the trick here is that you need to implement all of them **recursively**. You should also try to be efficient. In particular, no function should ever need to access any element of the list more than once. (i.e, don't go through the entire list once to find the max, and again to find the min). (Hint: if it's too difficult to solve a problem, think of a simpler problem you CAN solve recursively... like maybe returning a tuple of the two smallest values)

What to Submit

As always, your code should not use `import`, `input` or `print` anywhere. Make sure your function and file names are exactly as specified in this handout.

In order to ensure that you're doing the code yourself, we will take away access to the built in `min` and `max` functions. You code should not be using these functions in any way. (You shouldn't need to use any built in functions). You should also not have any loops anywhere in your code.

[1] This can be equal to the smallest number in the list, e.g., `second_smallest([1, 1, 3])` should return 1