CSCA48 Exercise 6

Due: March 3, 2017. 5:00pm

Last week we worked on recursion, so let's kick it up a notch!

More Complicated Recursion

In a file called `ex6.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

Wait a second... isn't this the same set of functions from last week? Almost... but here's the difference. We're no longer working with just lists of integers. We're working with **nested** lists of integers.

In particular, the lists for all four functions can be defined as lists of `L`s where `L` is defined as either being an integer, the empty list, or a list of `L`s.

Some example lists would be:

```
[1, 2, 3]
[1, [2, 3]]
[[1], [2, [3]], []]
```

As with the prevoius exercise, you must implement these functions `recursively`, you may not use loops anywhere in your code and you cannot rely on any built-in functions[2]. You should also be following the same rules of efficiency (you shouldn't pass through any list twice).

If you really planned your previous exercise properly, this shouldn't be a massive change, but make sure you really understand the problem before you start writing any code[3]. Remember, algorithm first, code second.

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.

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

[2] Except possibly for `isinstance`... hint hint...

[3] Remember that if you struggled with the previous exercise, you can always go to practicals to get help from the TAs