



*Assignment 15:*  
*Maximum Subarray Sum*  
**Due in class Tuesday, 1/22**

January 18, 2019  
CS: DS&A  
PROOF SCHOOL

This is a coding assignment. Since I'm trying to evaluate your code in a more automated way, I'm going to ask you to follow some submission rules *exactly*. Thank you!

1. Submit via Dropbox only. (I'm still experimenting with the best way to have you all submit, but let's try everyone using Dropbox this time. Whether we continue to use it depends on whether I can easily import files from your Dropbox folders as Python modules.)
2. Give your file the name `assignment_n.py`, where `n` is the assignment number. So this week's file is `assignment_15.py`. That's with one underscore.
3. Give your functions the exact names I ask for.
4. Any code that's not a part of the function must be placed in a `if __name__=="__main__"` block at the end, so that it doesn't run when I import it.
5. Remove all debugging print statements from your code before submitting!

**Problem 1.** Write a function called `max_subarray_1` that implements the  $\Theta(n \log n)$  divide-and-conquer algorithm from class for finding the max subarray sum. Your function should take a list as input and should not alter it. (Note that given input `[-2, -3]`, the answer should be 0, corresponding to an empty subarray.)

**Problem 2.** Write a function called `max_subarray_2` that uses dynamic programming to solve the same problem in  $\Theta(n)$  time. You may take either a top-down or bottom-up approach. For a review of dynamic programming, see the course notes from Block 2, and reread Assignment 11. The suggestion I gave in that assignment for solving the longest increasing subsequence problem is very relevant here!

**Problem 3.** If you weren't in class this Thursday (1/17), read through the course notes carefully to see what you missed, and *email me a question that you have from that material by EOD Monday*. You don't have to email me a question if you were in class, but you're welcome to.

Have fun!