CS 519-005, Algorithms (MS/MEng-level), Fall 2016
HW2 - Divide-n-conquer: mergesort, number of inversions, longest path

Due electronically on Canvas on Sunday Oct 2, 11:59pm.
No late submission will be accepted.

Include in your submission: report.txt, msort.py, inversions.py, longest.py.
DO _NOT_ ZIP YOUR SUBMISSION.
longest.py will be graded for correctness (1%).

Textbooks for References:
[1] CLRS

0. Which of the following sorting algorithms are (or can be made) stable?
   (a) mergesort
   (b) quicksort with the first element as pivot
   (c) quicksort with randomized pivot
   (d) selection sort
   (e) insertion sort
   (f) heap sort --- not covered yet (see CLRS Ch. 6)

1. Implement mergesort.

```
   >>> mergesort([4, 2, 5, 1, 6, 3])
   [1, 2, 3, 4, 5, 6]
```

   Filename: msort.py

2. Calculate the number of inversions in a list.

```
   >>> num_inversions([4, 1, 3, 2])
   4
   >>> num_inversions([2, 4, 1, 3])
   3
```

   Filename: inversions.py
   Must run in O(nlogn) time.

3. [WILL BE GRADED]

   Length of the longest path in a binary tree (number of edges).

   We will use the "buggy qsort" representation of binary trees:
   [left_subtree, root, right_subtree]

```
   >>> longest([[], 1, []])
   0

   >>> longest([[[], 1, []], 2, [[], 3, []]])
   2

   >>> longest([[[[], 1, []], 2, [[], 3, []]], 4, [[[], 5, []], 6, [[], 7, [[], 9, []]]]])
   5
```

   Note the answer is 5 because the longest path is 1-2-4-6-7-9.

   Filename: longest.py
   Must run in O(n) time.

Debriefing (required!): --------------------------

1. Approximately how many hours did you spend on this assignment?
2. Would you rate it as easy, moderate, or difficult?
3. Did you work on it mostly alone, or mostly with other people?
   Note you are encouraged to discuss with your classmates,
   but each students should submit his/her own code.
4. How deeply do you feel you understand the material it covers (0%–100%)?
5. Any other comments?

```
This section is intended to help us calibrate the homework assignments.
Your answers to this section will *not* affect your grade; however, skipping it
will certainly do.
```