Advanced Computer Science - Assignment #2

**Due date: Monday July 20 at 11:59PM**

1. **Cyclic Towers of Hanoi**
   1. We should be able to call your function with towers(n\_disks, start, stop, other), where each of the arguments is a number.
   2. Remember, in cyclic towers of Hanoi, peg one can only move to peg two, peg two can move only to peg three, and peg three can move only to peg one.
   3. Your program should print a list of moves in a format such as the following:  
        
       Move 1: Move from tower 1 to tower 2.  
       Move 2: Move from tower 2 to tower 3.  
       Move 3: Move from tower 1 to tower 2.  
       ...
   4. Please put this in its own file with a name that lets us know which problem it's solving.
2. **Alphabetical Path**
   1. For background on the puzzle, please read through [this page](https://www.brainbashers.com/abcpathhelp.asp) (and possibly play through one of the puzzles on your own).
   2. We've already written some functions for you and given some structure to the problem. For example, the constraints can be given as a string starting from the top left and then going clockwise. Thus, the linked example would have constraints specified by "TXYNFEJOQCH.....". The constraints will be put into a array similar to below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T** | **X** | **Y** | **N** | **F** | **E** | **J** |
| **V** |  |  |  |  |  | **O** |
| **R** |  |  |  |  |  | **Q** |
| **L** |  |  |  |  |  | **C** |
| **K** |  |  |  |  |  | **H** |
| **G** |  |  |  |  |  | **I** |
| **P** | **W** | **U** | **S** | **D** | **B** | **M** |

* 1. Your task is to fill the array given (a) a string of constraints, and (b) the starting position of "A". And, once solved, print the entire board (you can use our provided function for this). We've left comments throughout the starter file to help you know what's going on in each function *or* what you may want to provide in some missing functions.
  2. Please make sure that this code is saved and submitted in its own file. You can see the provided code in abc\_path\_starter.py.

1. **Sort Times**
   1. Suppose we have a method, unlucky\_quicksort, which -- unluckily -- keeps picking the largest element of the array it's sorting as its pivot. Using notation similar to what we used in class (for example, quicksort could be defined as ), prove that the running time of unlucky\_quicksort is for an array of length .
   2. merge\_sort operates in the following way:
      1. Break the list into groups of 2
      2. Sort each group of two
      3. **Merge** neighboring groups of 2 into groups of 4 by comparing the smallest unmerged elements in both groups until all elements have been merged
      4. Continue as above (with increased group size, of course) until complete

Let's assume for convenience that we have a list of size where (and is an integer greater than 2). Using any method you'd like, prove that the running time of merge\_sort is .  
  
If you're unsure of how merge\_sort works, feel free to search for some more information or just check [this demonstration](https://blog.digilentinc.com/wp-content/uploads/2015/01/mergesort.gif).

* 1. Please submit either a .docx or a .pdf file containing your answers to these questions. If you're writing in Word, it might be useful to know that [ALT]+[=] initializes the equation editor.

1. **Senators and Representatives**
   1. Here's a few random (valid) addresses in the US:
      1. 46855 Lyndon Ave, Canton, MI 48187
      2. 123 Winter St, Waltham, MA 02451
      3. 22362 Cupertino Rd, Cupertino, CA 95014
      4. 6510 SE Lake Cir Dr, Stuart, FL 34997
      5. 601 E 3rd St, Austin, TX 78701
   2. Write a Python script (perhaps using libraries like requests, pyquery, or beautifulsoup) that can take a list of addresses (such as the ones above) and prints out information similar to the following for each address:  
        
      Senators for 46855 Lyndon Ave, Canton, MI 48187:  
      Johnny Senator  
      Sandra Senator  
        
      Representative(s) for 46855 Lyndon Ave, Canton, MI 18187:  
      Richard Representative
   3. Please put this in its own file with a name that lets us know which problem it's solving.