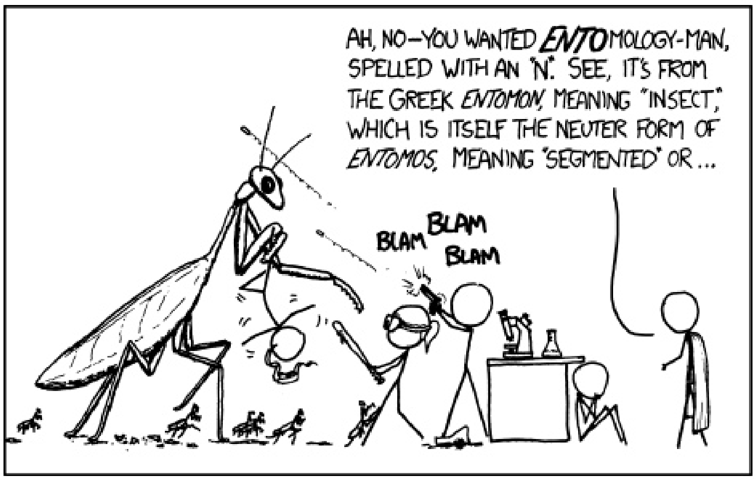
CS140 - Assignment 3

Due: Sunday, Feb. 11 at 10pm



http://www.xkcd.com/1012/

You must work with a partner on this assignment, but it can be anyone in the class, including someone that you've previously worked with. If you need a partner, please reach out ASAP.

1. **[25 points]** Stock Market Problem - the code

   Implement your algorithm from the previous assignment for the stock market problem in Python, C, or Java (ask me first if you want to use another language).

   Your submissions should be called `stockmarket.c`, `stockmarket.py`, or `StockMarket.java` if you're using C, Python, or Java, respectively. We will use an autograder that relies on this naming convention.

   Your code must allow the user to specify two filenames on the command line, say `infile.txt` and `outfile.txt` (these could be any filename, but I'm using these for examples.) We don't want to edit your code, so they need to be command-line parameters. For example:

   - If using C, your code might be compiled and executed as follows:

     ```
     % gcc stockmarket.c -o stockmarket
     % ./stockmarket infile.txt outfile.txt
     ```

   - If using Python 3[1], your code might be executed as follows:

     ```
     % python3 stockmarket.py infile.txt outfile.txt
     ```

[1] See the `https://docs.python.org/3/howto/argparse.html` module.

- If using Java[2], your code might be compiled and executed as follows:

```
% javac StockMarket.java
% java StockMarket infile.txt outfile.txt
```

The contents of the input file will be in the following form:

```
8
42
40
45
45
44
43
50
49
```

The first line specifies the number of days (which may not always be a power of 2). The following lines give the value of the stock on each day.

In this case the output file should contain the following:

```
3
2
40
45
45
```

The first line gives the length of the longest non-decreasing subsequence, the second line gives the day on which the subsequence begins (note the 1-based indexing!), and the rest of the lines give the prices of the stock on the days included in the subsequence. If there is more than one longest non-decreasimg subsequence, your code can return either one.

Most of the grade for this problem will be based on correctly implementing a divide-and-conquer algorithm that solves the problem described in last week's assignment. Note that we will evaluate outputs using `diff` so please make sure your output is in the format described above! We may also consider code efficiency in an absolute (wall clock) sense.

(Small) sample input and output files are on the course webpage.

2. **[25 points]** More Fun with Binary Search Trees

   (a) **[5 points]** Is the operation of deletion "commutative" in that deleting x and then y from a binary search tree always leaves the same tree as deleting y and then x? Argue why it is or give a counterexample. *Hint:* There are three different cases for deleting in a binary tree. Make sure you think about all of them. (We'll use the algorithm discussed in class, i.e., if a node has two children, it is replaced with its successor.)

[2] See `https://docs.oracle.com/javase/tutorial/essential/environment/cmdLineArgs.html`

(b) **[20 points]** Balanced?

Red-black trees maintain a set of five properties; in class we showed that maintaining those properties guarantees that the height of a red-black tree with n nodes is never more than $2\log(n+1)$. Consider another balanced binary search tree which maintains the following invariant: for any node x, the heights of the left and right subtrees of x differ by at most 1. We'll call these 1off trees.

i. [14 points] Prove by strong induction on the height of the tree that a 1off tree with height h has at least $f(h)$ nodes, where $f(h)$ is the hth Fibonacci number. (Recall that $f(n) = f(n-1) + f(n-2)$ and that $f(0) = f(1) = 1$.)

ii. [3 points] Use the previous part to show that a 1off tree with height h has at least $2f(h-2)$ nodes.

iii. [3 points] Finally show that a 1off tree with n nodes has $O(\log n)$ height.

Submitting

On Gradescope, you'll see two separate entries for this assignment (Assignment 3.1 for problem 1 and Assignment 3.2 for problem 2). For problem 1, submit your code as a single file with the appropriate extension. For problem 2, your answer should be a pdf, like you usually submit.