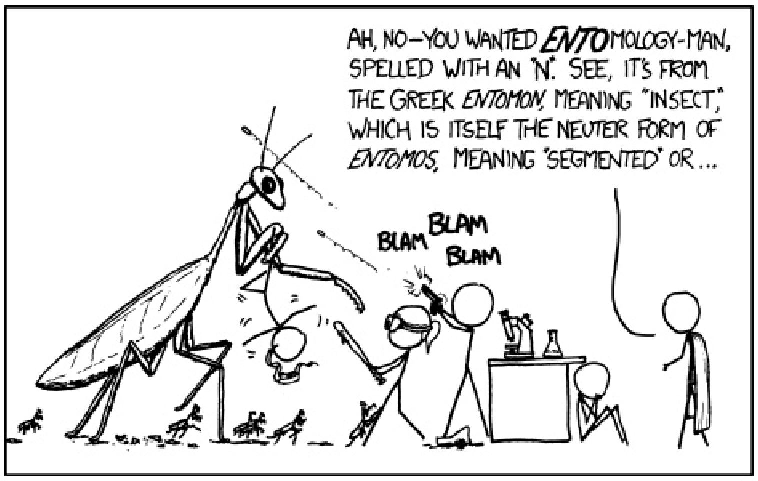
CS140 - Assignment 4

Due: Sunday, **Oct. 2 at 11:59pm**


AH, NO—YOU WANTED ENTOMOLOGY-MAN, SPELLED WITH AN "N." SEE, IT'S FROM THE GREEK ENTOMON, MEANING "INSECT," WHICH IS ITSELF THE NEUTER FORM OF ENTOMOS, MEANING "SEGMENTED" OR ...
BLAM
BLAM
BLAM

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

Note: For this assignment, you will have a week and a half (assuming you start after the checkpoint).

You may (and are encouraged) to work with a partner on this assignment, but you do not have to work with a partner.

Looking ahead: For the next assignment (Assignment 5) if you decide you want to work with a partner I will ask for you to work with someone that you haven't previously worked with.

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 the professor first if you want to use another language).

   Your code must allow the user to specify two filenames, say `infile.txt` and `outfile.txt` on the command line (we don't want to edit your code). 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:

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

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

- 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. [**20 points**] Hashtable fun

   (a) [**4 points**] Show the result of inserting 5, 28, 19, 15, 20, 10, 33, 12, 17 into a hashtable with collision resolution by chaining. The table should have 9 slots and use $h(k) = k \bmod 9$ for the hash function.

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

(b) [**4 points**] Show the result of inserting the first 6 of these into another hashtable using open addressing and linear probing.

(c) [**2 points**] For the these insertions, what was the largest number of collisions you had before finding an open slot and what key was it?

(d) [**10 points**] Open addressed hashtables can fill up. If you want to be able to support hashtables that are not of a fixed size, when an insertion happens and the table is full (or, more often, something like half full) a common approach is to create a new larger hashtable and insert the existing keys into this larger hashtable.

   i. Can you simply copy the entries from the old hashtable into the new hashtable? If so, justify why this still results in a correctly functioning hashtable. If not, describe what the problem is.

   ii. Argue that increasing the table size by a constant amount when it fills up will end up with a insertion being amortized $\Omega(n)$.

On Gradescope, you'll see two separate entries for this assignment (Assignment 4.1 for problem 1 and Assignment 4.2 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.