Assignment 4

csci2200, Algorithms

Honor code: *Work on this assignment alone or with one partner. Between different teams, collaboration is at level 1 [verbal collaboration only]. There are lots of resources online, such as animations, visualizations, practice problems, videos, and solutions— which you are encouraged to explore to deepen your understanding. However, you must be careful not to search for the specific problems in the assignment with the intent of getting hints for the solutions. Searching for the assignment problems on the internet violates academic honesty for this class.*

---

1. **Breaking eggs:**[1] Suppose you have an n-stories high building, and a bunch of eggs. An egg has a certain level l at which , if thrown from any level $\geq l$, it breaks. For example, an egg might have $l = 7$ meaning you can safely throw the egg down from levels 1 through 6, and it will not break; but if you through the egg from a level 7 or higher, it breaks.

   You are given a building and a bunch of eggs (all identical) and your goal is to find out the level l of the eggs. While you think about the problem, you can assume $n = 100$ (i.e. 100-level high building). But describe your solutions in terms of n

   (a) Describe an approach that only breaks one egg to find out l. How many throws does it do?

   What we expect: Explain the rationale of the algorithm and give pseudocode. Its analysis as function of n.

   (b) Describe an approach that minimizes the number of throws. How many eggs might it break?

   What we expect: Explain the rationale of the algorithm and give pseudocode. Its analysis as function of n.

   (c) Assume now you have two eggs. Describe an approach that minimizes the number of throws.

   What we expect: Explain the rationale of the algorithm and give pseudocode. Its analysis as function of n.

[1] This is from the textbook by Kleinberg-Tardo; also reported as an interview question by an alum

2. **Stoogesort:** One of your colleagues at work has proposed the following sorting algorithm, and your task is to evaluate it.

```
STOOGE-SORT(A, i, j)
if A[i] > A[j]: swap A[i] ↔ A[j]
if i + 1 ≥ j: return
k ← ⌊(j − i + 1)/3⌋
STOOGE-SORT(A, i, j − k)
STOOGE-SORT(A, i + k, j)
STOOGE-SORT(A, i, j − k)
```

   (a) Correctness:

do not turn in Work through an example and argue briefly that STOOGE-SORT correctly sorts any array of one element.

do not turn in Work through an example and argue briefly that STOOGE-SORT correctly sorts any array of two elements.

do not turn in Consider the algorithm but with the first line (that swaps elements $A[i]$ and $A[j]$) missing. Argue that it would not correctly sort by showing a simple counter-example.

do not turn in Work through an example array of 3 elements and see how it is getting sorted by STOOGE-SORT.

   i. Consider the state of the array A after the first recursive call finished and before starting the second recursive call (and assume the the recursive call correctly sorts). Consider the largest $n/3$ elements in A. Where might they reside? Make a claim and argue (briefly) why it's correct.
   What we expect: Claim: ..... Argument: ...

   ii. Consider the state of the array A after the second recursive call finished and before starting the third recursive call (and assume the recursive calls sort correcty). Consider the largest $n/3$ elements in A. Where might they reside? Make a claim and argue (briefly) why it's correct.
   What we expect: Statement: ..... Argument: ...

   (b) Running time: Give a recurrence for the worst-case running time of STOOGE-SORT and a tight asymptotic (Θ-notation) bound on the worst-case running time.

   What we expect: The recurrence, illustrate the process to find its solution, and its solution.

Evaluation

The assignment will be evaluated along several criteria:

1. **Correctness**: Is your solution correct?
2. **Justification**: Is your answer justified?
3. **Style**: Does it look professional and neat? Is the explanation written carefully in complete sentences, and well-organized logic? Is it easily human-readable? Is it easy to understand?

    - Assignments should be typed. Feel free to annotate the pdf to add figures and formulas which are too time-consuming to type.
    - Write each problem on a separate page or leave plenty of space between problems so that we can write comments.
    - Try to put yourself in the position of the reader. If you hadn't been thinking of this problem for 3 hours, would your answers make sense to you?
    - Try to finish the assignment early, then step away for a day or two, and then come back to it and read it again. Chances are you'll find something you can write more clearly.
    - Look at posted solutions for style advice (if solutions are not posted, ask).