Homework 1

Algorithms, Spring 2023

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. **Finding the 1's in a Matrix:** Consider a 2D-array (or: matrix) A and suppose that each row in A consists of 1's and 0's such that, in any row all the 1's come before any 0's. The problem is to find the row of A that contains the most 1's. Denote the number of rows and columns in A by n (and assume they are equal).

   (a) Describe the straightforward $O(n^2)$ algorithm for this problem.
   We expect: Pseudocode and analysis and argue that it is correct, i.e. it finds the row with the largest nubmber of 1's. In general we want to explain the idea of the algorithm at a high level, but when an algorithm is straightforwrad that's not necessary and pseudocode is sufficient.

   (b) Describe an improved algorithm running in $O(n)$ time.
   We expect: Pseudocode and a brief English description of the idea of your algorithm.

   (c) Let $n = 10$; show an array that triggers worst case for your algorithm.

   (d) Same, but show an array that triggers best case for your algorithm.

   (e) Argue why your algorithm from (b) runs in $O(n)$ time.

   (f) (Optional) Argue that your algorithm from (b) is correct, i.e. it finds the row with the largest number of 1's. *Hint: Try to formulate a loop invariant and prove it by induction on the iteration i of the algorithm.*

2. **Finding min and max:** Describe an algorithm for finding both the minimum and the maximum of n numbers with fewer than $3n/2$ comparisons in total.

Hint: Start by counting how many comparisons it takes to find the min and the max, and go from there.

We expect: Pseudocode and a brief English description of the idea of your algorithm.

(b) Analysis: Analysize your algorithm and show that it performs fewer than $3n/2$comparisons.

(c) Correctness: Give a brief justification that your algorithm is correct, i.e. it correctly finds the min and max.

3. **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 when you describe your solutions below, do it in terms of n.

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

   We expect: The rationale of the algorithm and pseudocode. Analysis function of n.

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

   We expect: the rationale of the algorithm and pseudocode. Analysis function of n.

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

   We expect: the rationale of the algorithm and pseudocode. Analysis function of n.

   (d) (Optional) For $n = 100$, minimize the number of throws to find the level. What's the best you can do?

   (e) (Optional) Assume now you have three eggs. Describe an approach that minimizes the number of throws.

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

Evaluation

The assignment will be evaluated along three general criteria:

1. **Correctness**: Does the algorithm solve the problem correctly?
2. **Analysis**: Is the running time of your algorithm analyzed? Is the algorithm efficient?
3. **Style**: An important goal of the assignments is to develop a good algorithmic writing style. Is the algorithm well described and high-level pseudocode included ? 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? detailed guidelines below[2]:

    - Assignments should be typed. Feel free to annotate the pdf to add figures and formulas which are too time-consuming to type. I recommend learning LaTex, but: some problems will require a lot of formatting (e.g. recurrences) which will be a time sink. In that case, simply annotate the equations on the pdf using the iPad.
    - Write each problem on a separate page (this leaves us space to write comments)
    - One of the skills you'll develop is how to communicate algorithms clearly. We expect your assignments to be clearly written and easily human-readable, with complete sentences and well-organized logic, and should definitely not be your first draft.
    - When the problem asks for an algorithm, please write *pseudocode*. Pseudocode should be clear enough that a student who took 1101 can understand what your algorithm is doing, and could implement it in a language of their choice, without thinking too hard. At the same time, pseudocode is **not actual code**, and should *not* include details that are straightforward and make the ideas too detailed/long and hard to follow. For e.g.it is preffered to say "find the max element in the array" (basic straightforward process) rather then spell it out. Check lecture notes for examples of pseudo-code.
    - It is good practice to include an English description of what your pseudocode is doing, to help out the reader.
    - 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).
    - Remember the study groups/office hours and come talk to us. The homeworks are not exams—they are not there to test you— they are there to create opportunities to learn. We expect everyone to do well on the homework, and to take the time to write carefully.

[2]Credit: inspired most recently by Stanford University, cs161