

Lab 1 Instructions

Assigned: 1/23/2019

Due Date: 2/8/2019

Part I

Compute the exact number of times that the statement `TICK` is executed in the following pseudocode fragments, as a function of n . Show your work, including for each loop the minimum and maximum value of the loop counter, as a function of n and/or the next outermost counter. Give your answer in closed form, with no remaining summations.

Reminder: what is “pseudocode”? It is a way of expressing algorithms that describes at a high level what they do but omits details specific to any particular language (Java, C, Python, etc.). You’ll find lots of pseudocode in your textbook. If you have any question about what the code blocks below mean, please ask.

The main thing that may be new to you in pseudocode is the use of \leftarrow to indicate assignment: “ $j \leftarrow 1$ ” would be written in Java as “`j = 1;`”. Pseudocode uses \leftarrow because it often uses “ $=$ ” to mean equality testing (where Java would use `==`).

1. **for** j **in** $0 \dots n$ **do**
 for k **in** $0 \dots j + 1$ **do**
 TICK
2. $i \leftarrow 1$
 while $i \leq n$ **do**
 TICK
 $j \leftarrow n$
 while $j > i$ **do**
 TICK
 $j--$
 $i++$
3. TICK
 $i \leftarrow 1$
 while $i \leq n$ **do**
 TICK
 $j \leftarrow 1$
 while $j \leq i \times i$ **do**
 TICK
 $j++$
 $i++$

Part II

Answer each of the following questions. Justify your answers using either the definitions of O , Ω , and Θ or the techniques shown in class (along with basic math).

4. Does $(n - 1)^2 = \Theta(n \log n)$?
5. Does $8^{\log n + \log \log n} = \Omega(n^3)$?
6. Does $n \log_7 n = \Theta(n \ln n)$?
7. Does $(3n - 10) = \Omega(\log^2 n)$?
8. Does $n^{16/15} = O(n \log n)$?
9. Let $f(n)$ and $g(n)$ be non-negative functions of n .
If $f(n) = O(g(n))$, does $f(n) + g(n) = \Theta(g(n))$?
10. Let $f(n)$ and $g(n)$ be non-negative functions of n .
If $f(n) = \omega(g(n))$, does $f(n)/g(n) = \omega(1)$?

Part III

Every night, Verizon has to update its database of every customer's telephone number. To enable fast lookups, the database includes an index that is a sorted list of all its phone numbers. This index must be rebuilt each night by re-sorting the contents of the database.

Verizon has hired you as a consultant to analyze their computing needs for the nightly sorting run. After carefully checking GitHub, you have found three sorting algorithms – A, B, and C – that can sort n phone numbers in $2 \times 10^{-7}n^2$, $10^{-5}n \log_2 n$, and $1.5 \times 10^{-4}n$ seconds respectively on a single processor.

11. What is the smallest problem size n_0 such that algorithm B is strictly faster than algorithm A for all $n \geq n_0$? (*Hint*: I don't know of an analytical solution to this problem, so try plugging in values for n or use Newton's method.) Justify your approach.
12. What is the smallest problem size n_1 for which algorithm C is strictly faster than algorithm B for all $n \geq n_1$? Justify your approach.
13. Describe how to construct a sorting algorithm that always achieves the best running time of any of algorithms A , B , or C for a given n .
14. Suppose the phone database contains 10^8 phone numbers. The updated information arrives at 4 AM, and the company demands that the new database be ready for use by 5 AM, i.e. one hour later. To meet this deadline, you may split the database evenly across k processors and sort each part separately. (For this problem, ignore the cost of putting the pieces back together.) How many processors do you need to meet your deadline with each of algorithms A , B , and C ?