Give full details of your argument.

Questions

1. (30 marks)

Suppose that you have to choose between three algorithms $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ for a given problem. The worst-case running time for each algorithm is given by

$$W_1(n) = 100n; \qquad W_2(n) = 2n \log_{10} n; \qquad W_3(n) = 0.1n^2.$$

Answer the following questions, explaining your assumptions and possible limitations of your analysis.

(a) Order the algorithms from fastest to slowest in the sense of asymptotic worst-case running time.

(b) Suppose that we run each algorithm on a large problem instance of size n. Then we feed in an input of size $100n$ and re-run the algorithms on the new input. What would you expect to happen to the running times, and why?

(c) Suppose that hardware limitations mean that your algorithm can only accept input of size at most 10^6. Which algorithm would you prefer, and why? Answer the same question with 10^6 replaced by 10^3.

(d) For which problem sizes is $\mathcal{A}_1$ the best algorithm to use? Answer the same question for $\mathcal{A}_2$ and then for $\mathcal{A}_3$.

(e) Suppose that we have 10^9 time units available. What is the maximum input size that can be processed by each of the algorithms in that time?

2. (20 marks)

Formally show that $0.1n + 10\sqrt{n}$ is not $O(\sqrt{n})$ **using the definition of O only**.

3. (25 marks)

Consider the obvious algorithm for checking whether a list of integers is sorted (with respect to the usual order): start at the beginning of the list, and scan along until we first find a successive pair of elements that is out of order. In that case, return false. If no such pair is found by the time we reach the end of the list, return true.

Our elementary operation is a comparison between two integers.

(a) What is the worst case running time of this algorithm on an input list containing n elements? Write all 5-element input lists which contain the numbers $1, 2, 3, 4, 5$ and yield the worst case.

(b) What is the best case running time of this algorithm on an input list with n elements? How many input lists are there which have size n, all elements being different, and yield the best case?

(c) Suppose that the input list is a random permutation of n distinct data items, and all such permutations are equally likely. Derive the average-case expected running time. Give both an exact and asymptotic answer.

4. (25 marks)

A certain algorithm has running time $T(n)$ on input of size n given by

$$T(n) = n\lfloor \lg n \rfloor + 2n - 2^{\lfloor \lg n \rfloor + 1}.$$

(a) Prove that $T(n)$ is $\Theta(n \log n)$.

(b) Prove that $T(n) - n \lg n$ is $O(n)$.

(c) Explain why the limit rule could not be used for (b).

Hint: Let $L(n) := \lg n - \lfloor \lg n \rfloor$ and think about the properties of the function L. For example, how large can it be, and what values does it take when $n = 1, 2, \dots$?

Submission

You should submit via Canvas, the following:

- A single PDF file containing your answers. Do the best you can with mathematical symbols. For exponents, write something like

  ```
  2^n
  ```

  if using plain text. Use LaTeX if you really want it to look good. Some WYSISYG word processors produce output that is probably acceptable for this assignment; some produce very poor output. A scanned handwritten submission is acceptable if and only if it is very neatly written.

  Any results from the textbook or lecture notes that you use must be clearly cited and all working should be shown.