Assignment 1 - Hints and Clarifications

COMP3121/9101 21T3

Released September 15, due September 29

This document provides some hints to help you solve the problems in Assignment 1. You are *not* required to follow these hints, and there may be alternate solutions which are equally correct.

Also included are the clarifications listed in the Assignment 1 FAQ on the Ed forum. Further clarifications may be added after this document is released.

General clarifications:

- **What data structures and algorithms can I use?**

  You can quote any data structure or algorithm from cs2521 (or equivalent), if you are using them in their generic form. If you need to modify the structure or algorithm, you must provide details of the modification. Also see this Ed post, which refers to:

  - sorting: merge sort, quicksort, bucket sort
  - searching: binary search
  - data structures: array, linked list, stack, queue, binary search tree, heap (priority queue), hash table.

- **Do the requested time complexities mean expected or worst case?**

  By default, we mean worst case. We will always explicitly specify if we want the expected time.

- **How should I format/explain my algorithms? Is pseudocode required? Are diagrams/examples allowed?**

  Explain the steps of your algorithm, and the logic behind these steps (when not obvious) in plain English if possible. Pseudocode can be used to supplement an answer, but in many cases it is less clear and harder to mark. Diagrams and examples are also fine to include, but again as a supplement; your solution should not rely on them.

  Solutions to the first exercise will be released in Week 2 which can provide some guidance, but keep in mind that these solutions are sometimes skeletal and your explanations should be thorough.

---

1. You are given an array A of n positive integers. A pair of indices (i, j) where $i < j$ is said to be *consistent* if $A[j] - A[i] = j - i$.

(a) (10 points) Design an algorithm which runs in *expected* $O(n)$ time and counts the number of pairs of consistent indices.

(b) (10 points) Design an algorithm which runs in *worst case* $O(n \log n)$ time and counts the number of pairs of consistent indices.

Clarifications: Nil.

Hint: Rearrange the given equation.

---

2. (20 points) You are given an array A of n positive integers, each no larger than m. You may assume that $m \leq n$. The *beauty* of an array is the fewest number of times that any integer from 1 to m inclusive appears in the array. For example, if $m = 3$ then the beauty of $\langle 1, 3, 1, 2, 3, 3, 2 \rangle$ is 2. However, if $m = 4$ the same array would have beauty 0.

   An index i is said to be *fulfilling* if $A[1..i]$ has strictly greater beauty than $A[1..i-1]$. Design an algorithm which runs in $O(n)$ and finds all fulfilling indices.

   Clarifications:

   - **What input is given in an instance of the problem?**
     The array A and the integer m.
   - **Does the array start from index 1?**
     Yes, in this course we will usually describe an array of length n as $A[1..n]$, so $A[1]$ is the first element. Note that $A[1..0]$ is an empty array.

   Hint: You do not need to test each index i in constant time, as long as the total time taken for all indices is linear.

---

3. You are given a string s of length n, constructed from an alphabet with k characters. You may assume that all k different characters appear at least once in s.

   A *substring* of s is a contiguous sequence of characters within s.

   Design an algorithm which runs in $O(n)$ time and finds the length of the shortest substring of s which contains all k different characters.

   A solution for the case $k = 3$ will earn up to 10 points.

   Clarifications:

   - **What input is given in an instance of the problem?**
     The string s and the integer k.
   - **Are these ASCII characters?**
     The characters are not specified to come from any particular set, but you can assume that each character can be identified and assigned an ID number in constant time.

Hint: For $k = 3$, what form does an optimal substring take?

Hint: For the general case, what information do you need in order to calculate the length of the shortest such substring starting at index i? What preprocessing would let you access this information easily, and what needs to be updated when you move to index $i + 1$?

---

4. (20 points) You have $2n + 1$ ants along a line. Their positions are described by the real numbers $x_{-n} < x_{-n+1} < \ldots < x_{n-1} < x_n$. You will place food at a point x on the line, and all ants will walk along the line to reach the food. Find the value of x which minimises the total distance walked by all ants.

   Clarifications:

   - **Do I need to provide reasoning?**
     You must give reasoning for your answer. This reasoning may consist of plain English explanations and if necessary, formal mathematical proofs.

   Hint: Are there any values of x which you can easily prove *do not* minimise the total distance? Drawing a number line will help.

---

5. Determine whether:

   (I) $f(n) = O(g(n))$,

   (II) $f(n) = \Omega(g(n))$, i.e. $g(n) = O(f(n))$,

   (III) both (I) and (II), i.e. $f(n) = \Theta(g(n))$, or

   (IV) neither (I) nor (II)

   for the following pairs of functions, and justify your answer. Note that log denotes the natural logarithm, with base e.

   (a) (6 points) $f(n) = n^{1+\log n}$; $\quad g(n) = n \log n$;

   (b) (8 points) $f(n) = n^{1+\frac{1}{2}\cos(\pi n)}$; $\quad g(n) = n$;

   (c) (6 points) $f(n) = \log_2 n^{\log(n \log n)}$; $\quad g(n) = (\log n)^2$.

   Clarifications:

   - **Do I need to provide working out?**
     Yes, you should give full working.

   Hint: You should attempt to simplify each expression before comparing them. Review the information on asymptotic notation and logarithm laws from Lecture 2.
