UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

ALGORITHMS AND DATA STRUCTURES

Tuesday 28 May 2002

09:30 to 11:00

Examiners:

Prof. K. Turner (External)

Prof. C. Stirling (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. Recall that the *DFT* of a vector of n elements $a = (a_0, \ldots, a_{n-1})$ is defined as

$$\mathrm{DFT}(a) = (y_0, \ldots, y_{n-1})$$

where $y_k = \sum_{j=0}^{n-1} a_j \omega^{kj}$, and the *inverse DFT* of $y = (y_0, \ldots, y_{n-1})$ as

$$\mathrm{DFT}^{-1}(y) = (a_0, \ldots, a_{n-1})$$

where $a_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k \omega^{-jk}$. Here and throughout the question, $\omega = \omega_n$ is a principal n th root of unity.

(a) Let $n \geq 4$ be a power of 2 . Write down values for ω^0, $\omega^{n/4}$, $\omega^{n/2}$, and ω^n. Make clear which value corresponds to which expression. [*4 marks*]

[**Note:** You do not have to prove that these values are correct, just to specify them. The correct values are among: 1, -1, i, $-i$, $(1+i)/\sqrt{2}$, $(-1+i)/\sqrt{2}$, $(-1-i)/\sqrt{2}$, $(1-i)/\sqrt{2}$.]

(b) Explain the connection between the functions DFT and DFT^{-1} and the operations of *evaluation* and *interpolation* of polynomials in a single variable. [*5 marks*]

(c) Let $A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$ and $B(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1 x + b_0$ be polynomials of degree $n-1$, in the "coefficient representation". Describe the *point-value* representation of polynomials. Show how, by transforming between the two representations, it is possible to multiply A and B using $O(n \log n)$ arithmetic operations. [*8 marks*]

[**Note:** You may assume the existence of algorithms for computing DFT and DFT^{-1} using $O(n \log n)$ arithmetic operations.]

(d) Suppose that the method of part (c) were to be adapted, in a naive way, to compute the quotient $A(x)/B(x)$, i.e., the polynomial $Q(x)$ satisfying $B(x)Q(x) = A(x)$.

i. By giving a simple counterexample (i.e., by specifying polynomials A and B) show that the result may not be defined.

ii. Even when the result $Q(x)$ *is* defined, the proposed algorithm may fail. Explain what could go wrong. [*8 marks*]

2. (a) Define the *median* of n (distinct) numbers. (Assume that n is odd.) [*4 marks*]

(b) Describe a simple method for locating the median of n numbers in $O(n \log n)$ time. Take "simple" to mean: "could be implemented in a shell script employing a handful of UNIX commands" or "could be implemented as a Java program invoking one or two standard methods". [*4 marks*]

(c) Describe a more sophisticated algorithm for finding the median of n numbers in worst-case time $O(n)$, that starts by dividing the n numbers into groups of five. (This algorithm in fact solves the more general problem of finding the "i th order statistic" of the n numbers.) [*10 marks*]

(d) Write down the recurrence governing the runtime $T(n)$ of the algorithm from part (c), explaining briefly how each of the terms in the recurrence arises. Show that $T(n) \leq cn$ satisfies the recurrence for some c and all sufficiently large n. [*7 marks*]

[**Note:** It does not matter if your estimates for the sizes of recursive subproblems are out by small additive terms, as long as the multiplicative factors are correct.]

3. (a) Explain what is meant by the *convex hull* of a set of n distinct points in the plane. [*4 marks*]

   (b) *Graham's scan* finds the convex hull of a set of n points in $O(n \log n)$ time. Give a clear description of the algorithm (in pseudocode or precise English). Justify the claimed runtime of the algorithm. [*12 marks*]

   [**Note:** You are not required to justify correctness of the algorithm.]

   (c) By means of an example (or more accurately a sequence of examples), demonstrate that Graham's scan may "pop" or "reject" k points in sequence, for k arbitrarily large. [*4 marks*]

   (d) Consider the problem: given a set of points $p_1, p_2, \ldots, p_n$ in the plane, and a further point q (distinct from $p_1, p_2, \ldots, p_n$), determine whether q lies in the convex hull of $\{p_1, p_2, \ldots, p_n\}$ (including the boundary of the convex hull). Describe a *simple* method — using the algorithm of part (b) as a subroutine — for solving this problem in time $O(n \log n)$. [*5 marks*]