

One and a half hours

**UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE**

Algorithms and Imperative Programming

Date: Thursday 19th January 2012

Time: 09:45 - 11:15

**Please answer any TWO Questions from the THREE questions provided
Use a SEPARATE answerbook for each Question**

**For full marks your answers should be concise as well as accurate
Marks will be awarded for reasoning and method as well as being correct.**

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are
not programmable and do not store text.

[PTO]

1. Algorithm design.

For each of the following computational tasks

- (i) describe an algorithm for the task. Your description may be a program in a standard language, in pseudocode, or a clear and precise step-by-step description. You should explain your algorithm. Marks are awarded for a correct algorithm. Some marks will also be awarded for efficiency: the more efficient your algorithm is, the more marks it will be awarded.
 - (ii) give the worst-case time complexity of your algorithm in terms of the size of the input and the number of operations required. Explain your answer.
- a) Remove duplicates from a list of integers, i.e. given a list of integers, return a list of integers whose elements occur only once in the result and are exactly the elements of the given list (the order of elements in the result list is not important). (6 marks)
 - b) Given a list of integers as input, determine whether or not two integers (not necessarily distinct) in the list have a **sum** k . For example, for list $[2, 10, 5, 3, 7, 4, 8]$ and $k = 17$, there is a pair, 10 and 7 such that $10 + 7 = 17$. (7 marks)
 - c) List symmetric difference: Given two lists of integers, compute a list of integers which consists of those integers which are in one or the other of the given lists, but *not* in both (the order of the result list does not matter and, if numbers appear several times in the lists, the symmetric difference need not reflect this multiplicity - though it may). For example, given lists $[2, 5, 3, 8, 2, 4, 7]$ and $[6, 7, 2, 4, 9, 1]$, one possible symmetric difference list is $[5, 3, 8, 6, 9, 1]$. (7 marks)

2. Sorting

- a) Consider a sorting algorithm for sorting a list of integer keys into ascending order. For the sorting algorithm to be correct, its output must possess two essential properties.

(Property 1) The list of keys must be in nondescending numerical order.

(Property 2) ?

What is the second essential property the output must have? (1 mark)

- b) Which of the following statements correctly describe Quicksort? (Indicate all that apply).

- A. It is a distribution-based sort
- B. It can be implemented as an in-place sort
- C. It uses divide-and-conquer
- D. It does not always terminate with a sorted list
- E. Its worst-case complexity is $O(n^2)$
- F. It is slow in practice on unsorted inputs

(3 marks)

- c) Specify an efficient and simple $O(n)$ algorithm to sort a list of items whose keys are binary digits i.e. every item has a key of either a 0 or 1. (Hint: it is sufficient to name the algorithm and a simple modification to it, to make it suitable for this case). (2 marks)

- d) A database consists of a number of Person data-items. Each Person data-item has a Name key and an Age key. The database has a *stable sorting* procedure built in, which can be used to sort by either of these two keys.

A user sorts the data alphabetically by Name and then sorts the resulting sequence again in ascending order by Age. Which of the columns of data (A, B or C) might be (part of) the output after these two sorting processes have been completed? Explain your answer.

(A):

Richards	22yrs
Smith	19
Smith	20
Smith	22

(B):

Smith	19yrs
Smith	20
Richards	22
Smith	22

(C):

Smith	19yrs
Smith	20
Smith	22
Richards	22

(2 marks)

- e) A certain divide-and-conquer algorithm is one ‘conquer’ step away from finishing a sorting process. Here is the current state of the sorting:

1,6,11,13,13,15,22,4,17,45,46,76,77

Which well-known algorithm could this be? Explain what is done in the final sorting step and state the complexity of that step.

(3 marks)

- f) Robert Sedgwick recommends using the median of the first, middle and last element as the pivot in Quicksort. Consider the following list of numbers

15, 3, 6, 22, 1, 2, 3, 4, 1

- i) If this “median-of-three” idea is used to select the pivot, how many numbers are less than, how many equal to, and how many greater than the selected pivot? (Hint: median means the middle value, when values are arranged in ascending order) (1 mark)

- ii) Different versions of Quicksort may process the numbers in a different order, and so select different pivot values. But which two numbers in the list above will certainly NOT be the pivot when the numbers greater than the pivot (from the last step) are divided again in the next step? (2 marks)

- g) A software engineer wants to write a program to sort arbitrary non-negative integers into an unusual order: the number 5 should be the smallest number, the number 0 should be the largest one, and all other numbers should be ordered normally.

E.g. the numbers 0,1,5,2,3,0,1,5,4,632,0,7 would be sorted to 5,5,1,1,2,3,4,7,632,0,0

She uses a sorting algorithm that calls a comparison function normally defined as follows:

```
int mycompar(int a, int b)
{
    if (a>b)
        return 1;
    if (a<b)
        return -1;
    if (a==b)
        return 0;
}
```

Keeping to the same return values for the relations (0 meaning $a==b$, 1 meaning $a>b$, -1 meaning $a<b$), rewrite the comparison function so that it would achieve the intended sort order when called by a comparison-based sorting algorithm.

(3 marks)

- h) What is the minimum number of pair-wise comparisons needed to sort four arbitrary integer keys (in the worst case)? Explain your answer with reference to a binary tree of comparisons. (3 marks)

3. Complexity

- a) Two functions $f(n)$ and $g(n)$ have complexity $O(1)$ and $O(n)$ respectively. Given this, calculate the *worst-case complexity* of the following code fragments. Give your answers in Big-Oh notation.

```
i) if(n < k)
    g(n);
else
    for(i=0; i<n; i++)
        g(i);
```

(2 marks)

- ii) In the following, the relationship between n and m is unknown

```
for(j=0; j<n; j++)
    f(n);
for(j=0; j<m; j++)
    f(m);
```

(2 marks)

- iii) In the following, the relationship between n and m is unknown

```
k=n;
while(k>0)
    g(n);
    k=k/2;    { Comment: this is integer division, so 1/2=0 }
end while;
for(j=0; j<m; j++)
    f(m);
```

(2 marks)

- b) Simplify your answers to a(ii) and a(iii), using the additional information that m is $O(\log n)$. (2 marks)

- c) A function of one variable runs in $O(\log_2 n)$ time. If I give it first an input of size m , and then an input of size $16m$, which of the following is true about the two running times? Note all that apply. (Assume m is a reasonably large integer)

- (A) The first running time is shorter than the second
 (B) The second running time will be a constant *times* the first
 (C) The second running time will *add* a constant to the first

(2 marks)

- d) A function $f(n)$ is $100n^2$. Give a function $g(n)$ of complexity $O(n^3)$ such that $g(n) < f(n)$ for $n < 50$ and $g(n) \geq f(n)$ for all $n \geq 50$. Show any working.

(2 marks)

- e) Jack has written an algorithm, and decides to estimate its complexity experimentally. He collects the following data from two runs of his algorithm.

Input size n	Observed running time $t(n)$
20	$100\mu s$
40	$190\mu s$

- i) Jack guesses the complexity is linear based on his data. Give a linear expression for $t(n)$ that fits the data. (2 marks)
- ii) You tell Jack that his method of data collection is not very good. Write down three things he should have done to make sure his data are informative of the true running time complexity. (3 marks)
- iii) Jack carries out your instructions, and collects better data. Explain concisely how he could work out an expression for the complexity of his algorithm from this extra data, assuming that the complexity of his algorithm is believed to be a polynomial in n . (3 marks)