Xi'an Jiaotong-Liverpool University 西交利物浦大学

PAPER CODE	EXAMINER	DEPARTMENT	TEL
INT102		Intelligent Science	

2nd SEMESTER 2020/21 EXAMINATIONS (RESIT)

BACHELOR DEGREE – Year 2

ALGORITHMIC FOUNDATIONS AND PROBLEM SOLVING

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

READ THE FOLLOWING CAREFULLY:

1. The paper consists of Part A and Part B. Answer all questions in both parts. Total marks available are 100. Marks for this examination account for 100% of the total credit for INT102.

2. In Part A, each of the questions comprises 5 statements, for which you should select the one most appropriate answer. Answer all questions in Part A using the Multiple Choice Answer Sheet. Please read the instructions on the Multiple Choice Answer Sheet carefully and use a HB pencil to mark the Multiple Choice Answer Sheet. If you change your mind, be sure to erase the mark you have made. You may then mark the alternative answer.

3. Answers to questions in Part B should be written in the answer booklet provided.

4. This is an OPEN BOOK examination. You can reference textbooks and notes but discuss with other students in any way is not allowed.

5. Answer all questions using the Multiple Choice Answer Sheet at the end of the paper. Please read the instructions on the Multiple Choice Answer Sheet carefully.

6. The time of the exam is strictly limited to 2 hours.

7. For students who take the exam online, at the end of the examination, be absolutely sure to submit your answer via Learning Mall. The time for submission of your answer via Learning Mall is strictly limited to 15 minutes. Once the time is over, the submission link will be closed.

8. All answers must be in English.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

PART A

MCQs

PART B

Question I (20 marks)

1. Briefly describe the idea of the divide-and-conquer technique. **3**

2. Given an array A of n numbers, $(n \geq 1)$

   a) Design a divide-and-conquer algorithm for finding values of both the largest and smallest elements in A. **6**

   b) Set up a recurrence relation for the number of key comparisons made by your algorithm and justify it briefly. **5**

   c) For $n=2^k$, solve the recurrence relation set up in b). **4**

   d) What is the worst case time complexity of your algorithm (in big-O notation)? **2**

END OF THE PAPER

2.(a).

```
Algorithm divide-and-conquer (A[0...n-1], l, r)
// Input: A[0...n-1], 0, n-1
// output: largest and smallest

if l=r then
    return (A[l], A[r])

mid = ⌊(l+r)/2⌋
(l1, l2) = divide-and-conquer (A[0...n-1], l, mid)
(l3, l4) = divide-and-conquer (A[0...n-1], mid+1, r)
    smallest = min (l1, l3)
    biggest = max (l2, l4)
    return (smallest, biggest)
```

(b). $T(n) = \begin{cases} 2T(\frac{n}{2}) + O(2) & n > 2 \\ O(1) & n \leq 2 \end{cases}$

(c). $T(n) = 2T(\frac{n}{2}) + O(2)$ $n=2^k$ $k=\log_2 n$

$= 2(2T(\frac{n}{2^2}) + O(2)) + O(2)$

$\ldots\ldots$

$= 2^k O(1) + O(1)(1+2+\ldots+2^k)$

$= n + \frac{1-2^k}{1-2} \cdot 2 = O(n)$

(d). $O(n)$