Index Number:							

UNIVERSITY OF MORATUWA

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BSc Engineering Honours Degree
2012 Intake Semester 4 Examination

CS2022: DATA STRUCTURES & ALGORITHMS

Time allowed: 2 Hours May 2015

ADDITIONAL MATERIAL: *None*

INSTRUCTIONS TO CANDIDATES:

1. This paper consists of **two (2) Sections** in **15** pages.
2. Answer **all** questions in the question paper itself **in the space provided**. If you make a mistake or need additional space you may attach additional sheets.
3. Section **A** contains **twenty-five (25) MCQ and/or short answer questions**. **Answer these questions in the space provided in the paper**.
4. **For each MCQ question** in section A, there is **only one correct answer** and you are expected to **clearly mark only one choice**. There is **no penalty** (negative marks) for wrong answers.
5. Each question in section A carries 2 marks.
6. Section B contains **two (2)** essay type questions.
7. The maximum attainable mark for each question in section B is given in brackets.
8. The maximum attainable mark for this paper is 100. This examination accounts for 60% of the module assessment.
9. This is a closed book examination.

   NB: It is an offence to be in possession of unauthorized material during the examination.

10. Only calculators approved and labelled by the Faculty of Engineering are permitted.
11. Assume reasonable values for any data not given in or with the examination paper. Clearly state such assumptions made on the script.
12. In case of any doubt as to the interpretation of the wording of a question, make suitable assumptions and clearly state them on the script.
13. This paper should be answered only in English.

SECTION A

1. Which of the following factors affect the running time of a program?

   I. Amount of processing required by the algorithm
   II. Size of the input dataset
   III. Nature of the input dataset

   a) I Only
   b) I and II only
   c) I and III only
   d) All three

2. Which of the following statements are true regarding evaluation of algorithms?

   I. Best-case running time of an algorithm is a very good representation of the efficiency of an algorithm.
   II. Space efficiency is an important characteristic in evaluating algorithms.
   III. If two algorithms have same worst case running time, their average case running time must be same.

   a) I Only
   b) II only
   c) I and III only
   d) II and III only

3. Following array is to be sorted using insertion sort.

25	12	44	55	22	66	33	77

   The array after first and second iteration of insertion sort are shown below.

   After First Iteration:

12	25	44	55	22	66	33	77

   After Second Iteration:

12	25	44	55	22	66	33	77

   What will be the array after the **fourth** Iteration of the Insertion sort?

4. Which of the following statements are true regarding the efficiency of sorting algorithms?

   I. Worst case time complexity of merge sort and quicksort are the same.
   II. Space complexity of merge sort is worse than that of heap sort.
   III. Worst case time complexity of quicksort and bubble sort are the same.

   a) I Only
   b) II Only
   c) I and II only
   d) II and III only

Continued …

5. If $f(x) = 8x^3 + 7x + 25$ and $g(x) = x^3 + 2x^2 + 4$ and $h(x) = 25x^2 + 25$ which of the following statements is **correct**?

   a) $f(x) \in \Theta(g(x))$ and $g(x) \in \omega(h(x))$ and $h(x) \in o(f(x))$
   b) $f(x) \in o(g(x))$ and $f(x) \in \Omega(g(x))$ and $f(x) \in \omega(h(x))$
   c) $g(x) \in O(h(x))$ and $g(x) \in \omega(h(x))$ and $g(x) \in \Omega(f(x))$
   d) $h(x) \in \Theta(f(x))$ and$g(x) \in \Omega(f(x))$ and $f(x) \in O(g(x))$

6. Which of the following statements is **not** correct?

   a) If $f(x) \in \Theta(g(x))$ and $g(x) \in \Theta(h(x))$, then $f(x) \in \Theta(h(x))$ is always true.
   b) If $f(x) \in O(g(x))$ and $g(x) \in \Omega\big(h(x)\big)$, then $f(x) \in \Theta(h(x))$ is always true.
   c) If $f(x) \in \omega(g(x))$ and $g(x) \in \omega(h(x))$, then $f(x) \in \Omega(h(x))$ is always true.
   d) All three of the above are correct.

7. Which of the following is not a divide and conquer algorithm?

   a) Binary search
   b) Heap sort
   c) Merge sort
   d) Quick sort

8. Which of the following statements are correct regarding asymptotic notation?

   I. If $f(x) \in \Theta(g(x))$, then $f(x)$ **cannot** be in the set $\omega(g(x))$
   II. If $f(x) \in o(g(x))$ and $g(x) \in O(h(x))$ then, $f(x) \in o(h(x))$ is **always** true
   III. If $f(x) \in \omega(g(x))$, then $g(x) \in \Theta(f(x))$ can **never** be true

   a) I Only
   b) I and II only
   c) I and III only
   d) All three

9. You are given following sorted array and asked to do a binary search for number 22. Write the sequence of numbers that you would compare before concluding that the number 22 is not in the array.

1	3	5	8	9	11	14	20	31	44	50

   ……..

10. If a binary search tree contains numbers 12, 27, 55, 33, 66 and 83, what will be the output of the in-order traversal of the tree? If the output cannot be determined, explain why it cannot be determined.

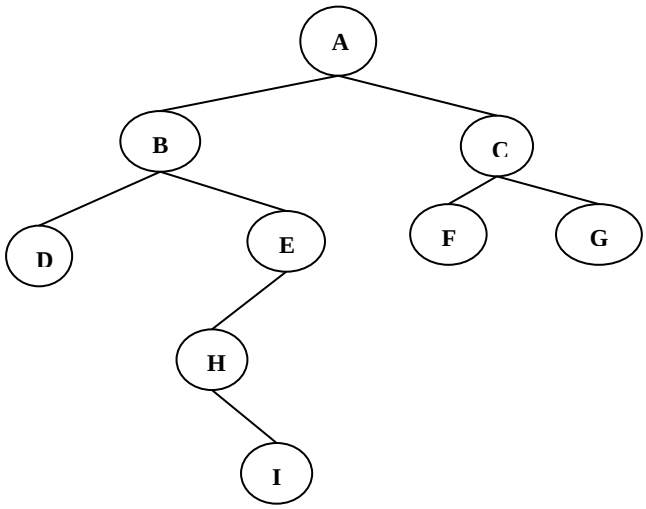
    ……..

    ……..

    ……..

Continued …

11. What is the result of pre-order traversal of the following tree?



……..

12. Draw a flowchart to express the algorithm to find the total of the elements in an array.

Continued …

Identify the most efficient data structure for each of the following scenarios in questions 13 to 16 and justify your answer in one or two sentences

13. To implement an algorithm to reverse any given string.

……

……

……

14. Support undo and redo operations of a simple notepad application.

……

……

……

15. Maintaining the list of actions which needs to be completed. The actions may come in any order and each action should be complete by the due date.

……

……

……

16. Maintaining list of printing tasks where each item has a priority associated with it. The printer will always print the task with highest priority and the priority of the tasks will change dynamically as they wait to be completed.

……

……

……

17. What is a Max-Heap?

……

……

……

Continued …

Consider the following scenario in answering questions 18 to 20.

You are required to store the index numbers, names and a phone number of the students of 12 batch of your department. The Index number of the students is to be used as the unique identifier for each record and it is of the form **120*dddX***, where *d* represents a digit and *X* is an uppercase letter. The application requires the records to be searched and accessed frequently. Two of your friends are arguing on which data structure to be used for this purpose. One friend is suggesting that a direct access table should be used and other friend is suggesting that a hash table should be used.

18. Briefly explain why a hash table is more suitable for this scenario.

……

……

……

……

19. Propose a suitable hashing function for this scenario. Explain your answer.

……

……

……

……

20. What would be a good size for the hash table? Explain your answer.

……

……

……

……

Continued …

21. Briefly explain a collision handing mechanism of a hash tables. Use a diagram to illustrate your answer.

22. Which of the following statements are correct regarding binary search trees?

    I. Worst case running time of a search in a binary search tree is O(*n*) where *n* is the number of nodes in the tree.
    II. Worst case running time of a search in a binary search tree is O(*h*) where *h* is the height of the tree.
    III. Balanced binary search trees can be efficiently represented using arrays.

a) I Only
b) II Only
c) II and III only
d) All three

23. What is the main difference between the divide and conquer approach and dynamic programming approach?

……

……

……

Continued …

24. What is greedy choice property?

……

……

……

25. Draw a suitable diagram to show how a tree with unbounded branching can be implemented.

Continued …

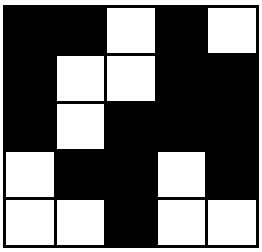
SECTION B

Q1. **[25 *marks*]**

(a) A black and white image is represented using a 2D-array in the following manner.

- 1 for places where we have a black dot (pixel)
- 0 for places where we have white dots

An object in an image is a set of black dots that are connected to each other horizontally or vertically. An example image with two objects and its array representation is shown below.



1	1	0	1	0
1	0	0	1	1
1	0	1	1	1
0	1	1	0	1
0	0	1	0	0

Write Pseudocode for a function that will erase all the dots (make them zero) that belong to one object. [6 *marks*]

```
void ErasePicture (int [][] imgArray, int imgHeight, int imgWidth,
int picRow, int picCol){
```

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

```
}
```

Continued …

(b) You are asked to implement two stacks in one array A[1…n] in such a way that neither stack overflows unless the total number of elements in both stacks together is n. Explain, using pseudo-code how PUSH (Stack, x), POP (Stack) operations are implemented in the given scenario (The PUSH and POP operations should run in $O(1)$ time). [8 *marks*]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Continued …

(c) Insertion sort can be expressed as a recursive procedure as follows. In order to sort an array A[1….*n*], we recursively sort the array A[1….(*n*-1)] and then insert the element A[*n*] into the sorted array.

i) What will be the terminating condition for the recursion? [2 *marks*]

……

……

……

ii) What will be the worst case time complexity of non-recursive part of the algorithm (inserting the element A[*n*] into the sorted array)? Explain your answer. [3 *marks*]

……

……

……

……

……

iii) Derive the recurrence formula to express the running time of the recursive insertion sort algorithm. [3 *marks*]

……

……

……

……

……

……

……

……

Continued …

iv) Solve the recurrence formula in part (iii) using a suitable method and express the worst case running time of the recursive insertion sort algorithm. [3 *marks*]

Q2. **[25 *marks*]**

(a)

i) Draw the binary search tree obtained by entering the following numbers in the same order. 75, 20, 100, 50, 10, 150, 5, 15, 22, 33, 44. [4 *marks*]

Continued …

ii) Show the resultant binary search tree obtained when number 100 is removed from the binary search tree of part (i). [2 *marks*]

iii) Show **all** the possible resultant binary search trees obtained when number 20 is removed from the binary search tree of part (ii). [6 *marks*]

Continued …

(b) An array of integers is shown below. Heap sort will be used to sort the array.

i. Show how the BUILD-MAX-HEAP(A) will convert the array into a heap. You are required to indicate each function call (each call to Max-Heapify) and the resultant array after that function call returns. [5 *marks*]

Function Call	**Resultant Array**							
Index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Initial Array	**55**	**33**	**88**	**77**	**111**	**22**	**44**	**66**

Continued …

ii. Show how the resultant array of part (i) will be sorted by the HEAP-SORT. You are expected to indicate each change to the heap (reducing the size of the heap, changing of an element in the heap, calling Max-Heapify). You have to only show how the three largest numbers are moved to the three last positions of the array. Showing of complete sorting process is not required. [8 *marks*]

Function Call	**Resultant Array**							
Index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Initial Heap								

--- ***End of the Paper*** ---