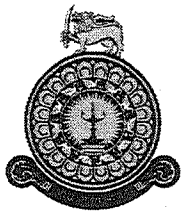
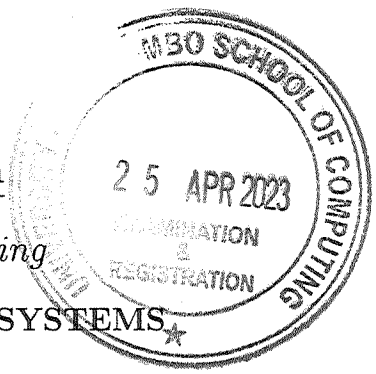


University of Colombo, Sri Lanka

University of Colombo School of Computing



BACHELOR OF SCIENCE IN INFORMATION SYSTEMS

First Year Examination - Semester II - UCSC AY 19 held in April 2023

IS 1114 - Data Structures and Algorithms I

Two (02) Hours

Answer All Questions

Instructions to Candidates

1. The medium of instruction and questions is in English.
2. Note that questions appear on both sides of the paper. If a page or a part of this question paper is not printed, please inform the supervisor immediately.
3. This paper has **two (02)** parts across **twelve (12) pages**.
4. MCQ should be marked on the MCQ answer sheet and the essay question on the answer book provided.
5. Write your **index number** clearly on all answer sheets.
6. This paper has two (02) parts, Part I and II. Part I consists of **thirty-five (35) MCQ** questions for **seventy (70) marks**, and Part II consists of an **essay question** for **thirty marks (30)**.
7. Calculators, Mobile Phones or any electronic devices capable of storing and retrieving data are **not** allowed.
8.

********** Multiple Choice Questions Marking Method **********

- Each MCQ question has five (05) answers with one (01) or more (up to 05) correct answers. Each MCQ question is worth two (02) marks. Suppose a question has two (02) correct answers and three (03) incorrect answers, *+2 marks will be shared between the correct options (1 mark per option), and -2 marks will be shared between incorrect options (-0.67)* per option. If one correct and incorrect answer is marked, the final mark for the question will be **1 + (-0.67) = 0.33**.
- Each question can have a positive or a negative value as the final mark. If a student scores less than zero (0) for an MCQ Question, zero (0) would be considered the mark for that question.

PART I

1. Which of the following statements is/are accurate regarding the array data structure?

   (a) All elements in an array can be accessed if the base address and the data type are known.

   (b) Arrays are contiguous in memory

   (c) Arrays are dynamically resizable structures.

   (d) Arrays are heterogeneous structures.

   (e) Arrays are homogeneous structures.

2. Which statements is/are accurate regarding a zero-based Array data structure of size five (5) with elements 6, 2, 3, 8 and 4?

   (a) A null pointer exception may occur if Array[5] is accessed.

   (b) If a delete operation is performed on the value two (2), the size reduces to four (4).

   (c) If Array[1] is accessed, value six (6) will be returned.

   (d) If the base address is added twice the data type, three (3) will be the element at the pointer.

   (e) If Array[5] is inserted with value nine (9), the size of the array would increase to six (6).

Consider the following pseudocode of **Algorithm A** and answer **Questions 03 to 06**.

```
AlgorithmA(int array[], int N, int k)
{
    for (i = 0 to N-k)
    {
            max = array[i];
            for (j = i to i + k -1)
            {
                if (array[j] > max)
                    max = array[j];
            }
            print max;
    }
}
```

PART I

3. What operation does **Algorithm A** perform?

   (a) Finding the array's maximum between zero (0) and the Nth index of the array.

   (b) Finding the maximum element in the array.

   (c) Finding the maximum element in the array between indexes N and k.

   (d) Finding the maximum of the moving block of size k starting for index zero (0) and move by one (1) each iteration.

   (e) Finding the maximum of the block of size k in the entire array.

4. If array 1, 2, 3, 4, 10, 6, 9, 8, 7, 5 was provided with k = 3, what would be the output of the above **Algorithm A**?

   (a) 10

   (b) 3

   (c) 3, 10, 9, 8

   (d) 3, 4, 10, 10, 10, 9, 9, 8

   (e) 3, 4, 10, 9, 8

5. How often would the inner loop execute if the input array is 9, 8, 6, 4, 3, 1 with k set to 4?

   (a) 2

   (b) 3

   (c) 4

   (d) 5

   (e) 6

6. What would be the output of **Algorithm A** if the k is set to one (1) for any given array?

   (a) A runtime error will occur

   (b) It will output ten (10).

   (c) It will output the entire input array.

   (d) There will be no output.

   (e) There will be unpredictable output for different arrays.

PART I

7. Which method is used by the array data structure to access its elements?

   (a) Direct
   (b) Exponential
   (c) Logarithmic
   (d) Random
   (e) Sequential

8. Which of the following is/are accurate regarding a stack data structure implementation?

   (a) Array-based stack implementation requires a unique pointer called the top of the stack.
   (b) Implementation of stacks requires paying attention to handling overflow and underflow errors.
   (c) LinkedList-based stacks are of finite size and have to be defined at initialization.
   (d) LinkedList-based stacks can use either the head or the tail pointer as the top of the stack.
   (e) Stacks can only be implemented using an array data structure.

Consider numbers 135 and 2976 added using stacks S1 and S2 and answer **Questions 9 and 10**.

9. What are the contents of the top of the stacks S1 and S2 after two (2) pop operations are performed on the stacks?

   (a) 1 2
   (b) 1 9
   (c) 3 7
   (d) 5 9
   (e) Stacks are empty

10. What is the top of the result stack after three (3) push operations?

    (a) 0
    (b) 1
    (c) 5
    (d) 7
    (e) Null value

PART I

11. Which of the following represents applications of the stack data structure?

    (a) Backtracking

    (b) Expressions conversion

    (c) Large number addition

    (d) Recursion

    (e) Resource allocation

12. Which statements regarding the queue data structure are true?

    (a) In a LinkedList implementation, if *addToHead* is considered an *enqueue* operation *deleteFromHead* must be the *dequeue* operation.

    (b) In a LinkedList implementation, if *addToHead* is considered an *enqueue* operation *deleteFromTail* must be the *dequeue* operation.

    (c) Queue is a first in first out data structure (FIFO).

    (d) Queue is a last-in-first-out data structure (LIFO).

    (e) Queues cannot be implemented using an array data structure.

Consider the numbers 34, 14, 27, 61, and 85 inserted into a LinkedList in the order given and answer **Questions 13 to 14**:

13. If the *addToHead* is used as the insert strategy, how many elements need to be visited before element 14 is visited?

    (a) 0

    (b) 1

    (c) 2

    (d) 3

    (e) 4

14. If the first two (2) elements are inserted using the *addToHead* strategy and the last three (3) elements using the *addToTail* method, how many elements must be visited before element 14 is visited

    (a) 0

    (b) 1

    (c) 2

    (d) 3

    (e) 4

PART I

15. Which of the following statements is/are accurate regarding Singly LinkedLists and Doubly LinkedLists?

    (a) Singly LinkedLists links are unidirectional, and Doubly LinkedLists are bidirectional.

    (b) Doubly LinkedLists outperform the Singly LinkedLists in search operations.

    (c) Doubly LinkedLists outperform the Singly LinkedLists in delete and insert operations.

    (d) If stacks were implemented using both types of LinkedLists, the Doubly LinkedLists variation would outperform the Singly LinkedLists variation.

    (e) Singly LinkedLists employ another pointer initialized one element before the main pointer to get the same effect of the Doubly LinkedLists in the delete operation.

16. What are stable-internal sorting algorithms?

    (a) Bubble sort

    (b) Insertion sort

    (c) Merge sort

    (d) Quick sort

    (e) Selection sort

17. Which of the following algorithms pays the least attention to order the elements in the input list?

    (a) Bubble sort

    (b) Insertion sort

    (c) Merge sort

    (d) Quick sort

    (e) Selection sort

18. Consider the situation in which assignment operation is very costly. Which sorting algorithms should be performed so that the number of assignment operations is minimized in general?

    (a) Bubble sort

    (b) Heap sort

    (c) Merge sort

    (d) Quick sort

    (e) Selection sort

PART I

19. What will be the output of the selection sort after the second pass when sorting the array 29, 10, 14, 37, 14?

    (a) 10, 14, 29, 37, 14

    (b) 10, 14, 37, 29, 14

    (c) 10, 14, 14, 29, 37

    (d) 14, 10, 29, 37, 14

    (e) 14, 29, 10, 37, 14

20. Which statement is correct about pivots if the array after the first partitioning using quicksort is 2, 5, 1, 7, 9, 12, 11, 10?

    (a) Could be 2

    (b) Could be 7

    (c) Could be 9

    (d) Could be 10

    (e) Could be 12

21. If the input is already sorted, what are the fair values for blanks stated as <blank> if it is in the best-case bubble sort algorithm?

```
boolean s = <blank>;
for(int j=arr.length-1; j>=0 && s; j=j-1)
{
        s = <blank>;
        for(int k=0; k<j; k++)
        {
                if(arr[k] > arr[k+1])
                {
                        int temp = arr[k];
                        arr[k] = arr[k+1];
                        arr[k+1] = temp;
                        s = <blank>;
                }
        }
}
```

    (a) False, False, True

    (b) True, False, False

    (c) True, True, True

    (d) True, False, True

    (e) True, True, False

PART I

22. What may be the output of the below code?

```
main (){
    int n, i;
    n=func(6);
    printf("%d",n);
}

func (int x){
    if(x==2)
        return 2;
    else{
            printf("+");
            func(x-1);
    }
}
```

(a) +++++2

(b) ++++2

(c) +++++

(d) 2

(e) +++2

23. What will be the error generated from the below code, and what should be the solution?

```
int something(int number) {
    if(number > 5)
        return 1;
    else
        return number * something(number-1);
}
something(4);
```

(a) Array index out of bound → if condition should be changed as (number <= 0)

(b) Stack overflow → if condition should be changed as (number <= 0)

(c) Stack underflow → if condition should be changed as (number == 0)

(d) Stack underflow → if condition should be changed as (number > 0)

(e) Stack overflow → if condition should be changed as (number == 0)

PART I

24. Which answer best describes the growth of the function **func1(x)** of the below code segment?

```
int func1(int x){
        if (x<1) return 1;
        else return ( func1(x-1) + func2(x) );
}

int func2(int x){
        if(x<2) return 2;
        else return ( func1(x-1) + func2(x/2) );
}
```

(a) Constant

(b) Cubic

(c) Exponential

(d) Linear

(e) Quadratic

25. What is a full binary tree?

(a) All the leaves are at the same level.

(b) Each node has exactly one or two children.

(c) Each node has exactly two children.

(d) Each node has exactly zero or one child.

(e) Each node has exactly zero or two children.

26. What is a complete binary tree?

(a) A binary tree is completely filled, except for the bottom level, which is filled from left to right.

(b) A binary tree is completely filled, except for the bottom level, which is filled from right to left.

(c) A tree In which all nodes have degree 2.

(d) Each node has exactly zero or two children.

(e) Each node has exactly two children.

PART I

27. Which of the following is/are the advantages of trees?

    (a) Efficient deletion of nodes

    (b) Faster Search

    (c) Hierarchical Structure

    (d) Path Selection

    (e) Undo/Redo Operations

28. Which of the following is/are accurate for binary trees?

    (a) Let T be a binary tree. For every $k \geq 0$, there are no more than 2k nodes in level k.

    (b) Let T be a binary tree with λ levels. Then T has no more than 2λ–1 nodes.

    (c) Let T be a binary tree with N nodes. Then the number of levels is at least $ceil(log(N+1))$.

    (d) Let T be a binary tree with N nodes. Then the number of levels is at least $floor(log(N+1))$.

    (e) Let T be a perfect binary tree with N nodes. Then the height of the tree is at least $ceil(log(N+1))$.

Consider a binary search tree with elements 40, 20, 65, 85, 10, 75, 30, and 80 inserted into the tree in the order specified and answer **Questions 29 to 35**.

29. What is the rightmost node of the left subtree?

    (a) 10

    (b) 20

    (c) 30

    (d) 80

    (e) 85

30. What is the leftmost node of the right subtree?

    (a) 30

    (b) 65

    (c) 75

    (d) 80

    (e) 85

PART I

31. What is the level difference between the right and left subtree?

    (a) 0
    (b) 1
    (c) 2
    (d) 3
    (e) 4

32. If node 40 is deleted, what are the possible candidates to take the position of the deleted nodes?

    (a) 10 and 85
    (b) 10 and 30
    (c) 20 and 85
    (d) 30 and 65
    (e) 30 and 75

33. If nodes 35 and 50 are inserted into the above-described tree and node 40 is deleted using the delete by merge technique, which of the following is/are accurate about the deletion and the resulting binary search tree?

    (a) 20 is a possible new root of the resulting binary search tree.
    (b) 65 is a possible new root of the resulting binary search tree.
    (c) The height of the resulting tree can be four (4).
    (d) The height of the resulting tree can be five (5).
    (e) The height of the resulting tree can be six (6).

34. Which of the following traversals are identified with specific names?

    (a) 10, 20, 30, 40, 65, 75, 80, 85
    (b) 10, 30, 20, 80, 75, 85, 65, 40
    (c) 40, 20, 10, 30, 65, 85, 75, 80
    (d) 40, 20, 65, 10, 30, 85, 75, 80
    (e) 85, 80, 75, 65, 40, 30, 20, 10

PART I

35. What is/are the height of the perfect binary tree for the binary search tree mentioned above once the node with value 40 is deleted?

    (a) 1
    (b) 2
    (c) 3
    (d) 4
    (e) 5

PART II

Answer Questions 1 and 2 in the answer book provided.

1. Write a pseudocode for the push and pop operations for a Linkedlist-based stack data structure. Your data structure should adhere to the following constraints. Clearly state your assumptions. **[20 marks]**

    - **Only one access pointer** for the LinkedList is possible (This LinkedList has **no head or tail pointers** and only has one pointer called TOS (top of stack))
    - You can decide on the structure of the LinkedList node. The pointers between nodes are not the access pointers (e.g. head/tail) mentioned above.
    - Push or Pop operation should not use additional pointers apart from **one temporary variable**.

2. Write a pseudocode to reverse the array's contents using **only one loop counter variable and one additional temporary variable**. Clearly state your assumptions. **[10 marks]**

——— *End of Paper* ———