# University of Colombo, Sri Lanka

*University of Colombo School of Computing*

## BACHELOR OF SCIENCE IN INFORMATION SYSTEMS

Second Year Examination - Semester II - UCSC AY20 [held in March 2024]

## IS2110 — Data Structures and Algorithms II

(Two (2) Hours)

**Answer ALL questions**



35

**Number of Pages 12** **Number of MCQs = 35; Essay Questions = 1**

**Important Instructions to candidates:**

I. This is a closed book exam.

II. 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.

III. This paper has two (02) parts, **Part I** and **Part II**.

IV. **Part I** consists of **thirty-five (35) MCQ** questions for **seventy (70) marks.**

V. **Part II** consists of an **essay question for thirty marks (30).**

VI. Answer ALL questions.

VII. Write your **index number** clearly on the answer sheets.

VIII. MCQ answers should be marked on the **MCQ answer sheet** and the answers to the essay question in Part II need to be written on the **Answer Book** provided.

IX. **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.

X. **Each question can have a positive or a negative value as the final mark.** Both negative and positive values would be added to calculate the MCQ section mark. If a student scores less than zero (0) for the MCQ section (Part I), zero (0) would be considered as the mark for that section (Part I)

XI. **Calculators, Mobile phones, Smart watches** or any other electronic device capable of storing and retrieving data are **not allowed.**

# PART I

1) How does the global balancing approach of the DSW algorithm, compared to local balancing strategies, impact the tree's performance for various operations?

(a) As the tree grows significantly, the complexity of global balancing algorithms decreases.

(b) Optimal for Dynamic Trees since there are many operations being performed.

(c) It may lead to temporary inefficiencies during updates, as the tree is not rebalanced immediately after each operation.

(d) It optimizes the time complexity of traversal operations by maintaining a more uniform tree structure.

(e) It can cause a delay in rebalancing, affecting the performance of operations until the global rebalance is triggered.

2) In the process of balancing a binary tree using the DSW algorithm, after transforming the tree into a right-leaning vine with 18 nodes, the initial step involves performing left rotations on the excess nodes to approach a perfectly balanced tree. How many left rotations are performed on the odd nodes in the vine during the initial step?

(a) 1 left rotation

(b) 2 left rotations

(c) 3 left rotations

(d) 4 left rotations

(e) 5 left rotations

3) After the initial left rotations for excess nodes in the DSW balancing phase, how is the process of balancing the remaining vine achieved?

(a) Completely rebuild the entire DSW tree from scratch to ensure perfect balance.

(b) Iteratively halve the size of the vine (rounded down to an integer). You then perform that number of left rotations on odd nodes within the vine.

(c) Perform a single left rotation on the root node of the entire tree.

(d) Repeatedly divide the size of the vine in half by performing left rotations on every other remaining node, starting from the leftmost node.

(e) The size of the vine is halved by performing left rotations on every node, starting from the root.

4) What are the correct steps to search for a key in a B+ Tree?

(a) Start at the root and linearly search each key in the node until the appropriate child pointer is found.

(b) Compare the search key with the keys in each node starting from the root, moving left for keys less than and right for keys greater than the compared key.

(c) If a key is not found in the internal node, the search is terminated as unsuccessful.

(d) Continue the search recursively or iteratively through the tree until the leaf level is reached.

(e) On reaching the appropriate leaf node, linearly search for the key within the node.

5) Which of the following are true regarding the deletion operation in B+ Trees?

(a) A key present only in a leaf node can be deleted directly if the node has more than the minimum number of keys.

(b) After deleting a key from an internal node, the empty space must be filled with the inorder successor from the leaf nodes.

(c) If the leaf node contains exactly minimum number of keys after deletion then merge the node with its immediate sibling.

(d) If the key to be deleted is present in the internal nodes and the deletion of a key from a leaf node causes underflow, then check to borrow a key from an adjacent leaf node through their common parent.

(e) In case of underflow, a leaf node always redistributes keys by merging with both immediate siblings to ensure maximum key occupancy.

6) B trees and B+ Trees are both designed for efficient storage and retrieval of data on secondary storage devices like disks. What are some key differences between them?

(a) B Trees can store data records (actual data) in both internal and leaf nodes, while B+ trees only store data records in leaf nodes.

(b) B+ trees are better suited for scenarios where frequent range queries are performed on the data.

(c) Data stored in a B+ tree can be accessed only directly.

(d) Height of a B+ tree is larger than the height of a B Tree for same number of nodes.

(e) To each node data access time is same for B Tree and B+ tree.

7) In a **B tree of order 4**, you have just deleted a key from a leaf node, and its immediate siblings contain a minimum number of keys. Considering the B Tree's properties and the need to maintain balance, what is the most likely next step in the process?

(a) Borrow a key from an adjacent leaf node's parent to balance for the key deletion

(b) A leaf node is merged with one of its siblings and a key from their shared parent.

(c) A leaf node merges with one of its immediate siblings.

(d) The parent node of the affected leaf is split to redistribute keys among its children, addressing the imbalance caused by the deletion.

(e) The B Tree does nothing further, accepting the imbalance as temporary until the next insertion operation.

8) Which strategy would **most likely** ensure the B Tree remains balanced and adheres to its properties after the deletion?

(a) If possible, the tree borrows a key from an adjacent sibling leaf node that has more than the minimum required number of keys.

(b) The tree immediately merges the leaf node with its parent node, regardless of the parent node's key count.

(c) The tree always splits the parent node to balance the tree.

(d) The leaf node is merged with one of its siblings, and if necessary, this merge propagates up the tree to maintain the B Tree properties.

(e) The tree replaces the deleted key with a placeholder value to maintain the minimum key count until the next insertion operation.

9) If a Jump search algorithm is implemented using a conventional (doubly) linked list, what will be the time complexity of such implementation?

(a) $O(\sqrt{n})$

(b) $O(n)$

(c) $O(1)$

(d) $O(nlogn)$

(f) $O(n^2)$

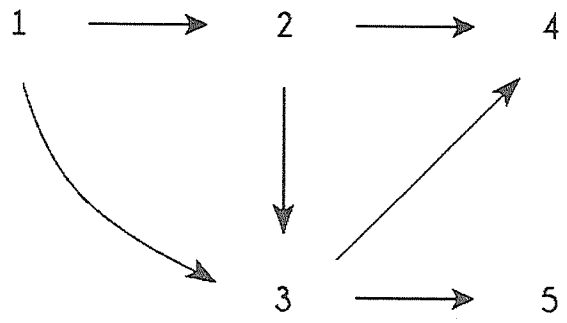
10) What is/are applications of topological sort?

(a) Building Makefiles in software projects.
(b) Deadlock Detection.
(c) Implementing Circular Queues.
(d) Sorting numbers stored in a two-dimensional array.
(e) Advanced-Packaging Tool (apt-get).

11) What is the ***running time*** of **Depth First Search** in a graph?

(a) O(V+E)
(b) O(V)
(c) O(log $V^2$)
(d) O(log V)
(e) O(V log V)

12) What is/are correct topological sort sequence(s) of the graph below.




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

13) The following function dynamically allocates memory for a new node, initializes its data field with the given value, and sets its next pointer to NULL. What line(s) can be used to complete the code.

```
struct Node* createNode(int data) {
  --------------------------------

  newNode->data = data;

  newNode->next = NULL;

  return newNode;

}
```

```
(a) int* newNode = calloc (1,sizeof(struct Node));
(b) struct Node* newNode = (struct Node*)malloc(1*sizeof(struct
    Node));
(c) struct Node* newNode = (struct Node*) calloc (1,sizeof(struct
    Node));

(d) newNode->data = NULL;

(e) newNode->data = newNode->next;
```

14) What is the time complexity of the code below?

```
public int calculate(int n)
    {
        int i, k = n;
        for (i = 1; i <= n; i=i*2) {
            System.out.println("Hello World !!!\n");
        }
    }
```

(a) O(n/2)
(b) O(n)
(c) $O(\log_2(n))$
(d) O(1)
(e) O(n*2)

15) *Over the last century, a great number of deterministic exact solving algorithms have been developed for the Hamiltonian cycle problem. Rooted in dynamic programming, the Bellman–Held–Karp algorithm is quite memory intensive, but by still holds the lowest time complexity of* $O(n^2.2^n)$.

What is type of time complexity this algorithm shows?

(a) Quadratic Time Complexity
(b) Linear Time Complexity
(c) Factorial Time Complexity
(d) Constant Time Complexity
(e) Exponential Time Complexity

16) What is/are the advantage(s) of **Adjacency Matrix Representation** of Graphs?

(a) Can be used to represent a graph in linear algebra.

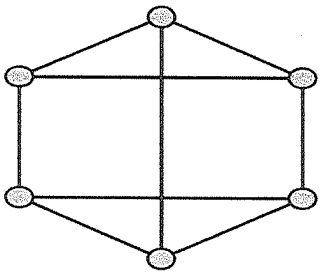
(a) Preferred when the graph is sparse.

(b) Graph traversal algorithms like DFS/BFS requires O(V + E) time.

(c) Constant time edge lookup.

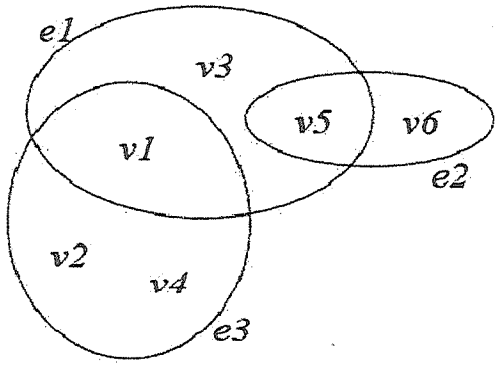
(d) Adding or removing edges from a graph is quick and easy.

17) What is the best explanation(s) for the graph given below?



(a) Tri-partite Graph
(b) 3-Regular graph
(c) Complete graph
(d) Multigraph
(e) Hypergraph

18) What is the best explanation(s) for the graph given below?




(a) Multigraph
(b) Hypergraph
(c) Complete graph
(d) Tri-partite Graph
(e) Regular Graph

19) Which of the following describes bipartite graph's properties?

(a) It has no odd cycles

(b) It can be colored with two colors

(c) It has an Eulerian cycle

(d) No two adjacent vertices belong to the same set

(e) Forms a universal sink on the graph

20) How many **spanning trees** can be derived from a complete graph which has 6 vertices?

(a) 7776
(b) 720
(c) 360
(d) 216
(e) 1296

21) A graph's complement (G′) is produced by

(a) Making a bi-partite graph
(b) Removing all edges from the original graph.
(c) Adding edges between non-adjacent vertices of the original graph and removing existing edges.
(d) Reversing the directions of all edges in the original graph.
(e) Cutting the graph's space complexity in half.

22) What is the purpose of the compression map functions in hashing?

(a) It transforms a key into an index within the hash table.
(b) It increases the size of the hash table dynamically.
(c) It compresses the size of the keys.
(d) It prevents collisions between keys.
(e) It converts non-numerical keys into numerical keys.

23) What could be a disadvantage of the division method in hashing?

(a) It is computationally intensive.
(b) It may result in clustering of hash codes.
(c) It requires a constant value for multiplication.
(d) It is not suitable for integer keys.
(e) Optimal for graph traversal.

24) In a **Radix Transformation hashing function** which uses base 8 and m=99, what is the output hash value for the key 235 (in base 10)?

(a) 235
(b) 99
(c) 56
(d) 12
(e) 3

25) In a **summing components - hash code map function** which uses ASCII table, what is the output for the following word:

CAT

(a) 72
(b) 216
(c) 65
(d) 67
(e) 211

26) A red-black tree with 63 internal nodes has height at most,

(a) 6
(b) 7
(c) 10
(d) 12
(e) 13

27) Which is/are application(s) of **Greedy algorithms**?

(a) Store Programming Language Keywords
(b) Travelling Salesman Problem
(c) Minimum Spanning Trees
(d) Cycle Detection in Graphs
(e) Fake Coin problem

28) As shown in the table below, there are 6 activities with corresponding start and end time, the objective is to compute an execution schedule having the maximum number of non-conflicting activities. What is/are possible **activity execution schedule(s)** using **greedy approach**?

| Start Time (s) | Finish Time (f) | Activity Name |
|---|---|---|
| 4 | 5 | a1 |
| 1 | 2 | a2 |
| 4 | 6 | a3 |
| 1 | 3 | a4 |
| 6 | 8 | a5 |
| 9 | 10 | a6 |

(a) a2, a3, a5, a6
(b) a2, a1, a5, a6
(c) a4, a1, a5, a6
(d) a4, a3, a6
(e) a4, a3, a5, a6

29) In the folding method of compression, what does the algorithm do with the key?

(a) It splits the key into smaller parts and adds them together.
(b) It multiplies the key with a constant value.
(c) It finds the square of the key.
(d) It rotates the key bitwise.
(e) Performs XOR operation on the key.

30) A **double hashing function** is defined as $h(k, i) = [h_1(k) + i*h_2(k)] \bmod m$, where

$h_1(k) = k \bmod 13$

$h_2(k) = [8 - (k \bmod 8)]$

$m=11$ and

$i=2$

Find the hash table position for the key 87.

(a) 0
(b) 1
(c) 4
(d) 7
(e) 11

31) Which algorithm(s) can be considered as implementation of the **divide and conquer approach**?

(a) Quick Sort
(b) Insertion Sort
(c) Topological Sort
(d) Binary Search
(e) Merge sort

32) In a red-black tree, what is the **maximum number of immediate red children** a black node can have?

(a) 1
(b) 2
(c) $\sqrt{n}$
(d) 8
(e) $\infty$

33) What is the time **average time complexity** of the interpolation search algorithm?

(a) O(n)
(b) O(log n)
(c) O(log log n)
(d) $O(n^2)$
(e) O(2n)

34) Based on the interpolation search, what will be the **calculated probe position** for the following array?

```
array = [2, 4, 7, 9, 12, 15, 17, 20, 33, 47]
```

Assumption: first index of the array is 1.

(a) 8
(b) 7
(c) 6
(d) 5
(e) 4

35) What is the purpose of the following algorithm?

```
int f1(int n){
    if (n <= 9)
      return n;
    else
    {
      a = n % 10;
      return f1(n / 10) + a;
    }
  }
```

(a) Summing all numbers between n and 9
(b) Summing all numbers between 1 and 9
(c) Summing all numbers between n/10 and 9
(d) Summing the digits of a number
(e) Summing all numbers between n/10 and n%10

## Part II

**Q1**

Consider the following directed graph.

```
G(V,E) where
V={A,B,C,D,E,F,G,H,I}
E={AB,AC,AD,BA,BE,BF,CA,CG,DA,DH,DI,EB,FB,GC,HD,ID}
```

a. Draw the resulting directed graph denoted by `G(V,E)`.

**(4 marks)**

b. Write a pseudocode for the **Bredth-First-Search** traversal algorithm using a **linked list**.

**(7 marks)**

c. On node `A`, perform Bredth-First-Search, and record the results in the proper sequence. Show intermediate steps. e.g. `BFS(G,A)`.

**(6 marks)**

d. Write a pseudocode for **Depth-First-Search** traversal algorithm using a **stack**.

**(7 marks)**

e. On node `A`, perform Depth-First-Search, and record the results in the proper sequence. Show intermediate steps. e.g. `DFS(G,A)`.

**(6 marks)**

*****