MBARARA UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATICS
End of Semester Two Examination for the Degree of Bachelor of Software Engineering

Course Code : SWE 1202
Course Name : Data Structures and Algorithms
Course Year : One
Academic Year : 2021 / 2022
Date : Monday 10th October 2022
Room : MLT
Duration : 1400 - 1700 HOURS

INSTRUCTIONS

1. The paper has two sections: A and B.
2. Section A is Compulsory & carries 40 marks. It consists of two parts:
   (a) Part I has multiple, true or false choice questions & carries 24 marks
   (b) Part II has problem solving questions and carries 16 marks
3. Attempt any **three** (3) out of the **four** (4) questions in section B. Each question carries 20 marks.
4. Begin each question of Section B on a new page.
5. This is not an Open Book Examination – Candidates should not consult any Reference Material During this sitting
6. No programmable Electronic device will be allowed in the examination
7. Any form of Examination Malpractice may lead to discontinuation from the University.
8. Do not write anything on the Question Paper
9. All answers and rough work (use the last page) should be in the answer booklet

This exam has 14 questions, for a total of 100 points

SECTION A

Part I :Multiple Choice Questions

Question 1: Iterators ..... **2 points**

(a) (1**mark(s)**) An iterator is:
A. a loop control variable B. an element in a collection. C. an interface D. an object that allows access to each element in a collection individually. E. a five syllable word

(b) (1**mark(s)**) The only method in the Iterable interface is:
A. remove B. iterator C. hasNext D. destroy E. create

Question 2: Searching and Sorting ..... **1 points**

(a) (1**mark(s)**) In a binary search, _____.
A. it is assumed that the search pool is large. B. it is assumed that the search pool is ordered. C. it is assumed that all of the elements are Strings. D. it is assumed that the search pool is unordered E. it is assumed that the search pool is small.

Question 3: General Trees ..... **7 points**

(a) (1**mark(s)**) A user has designed an interface for a binary tree abstract data type (ADT). Which method below requires knowledge of the purpose and organisation of the binary tree in order to design an implementation? A. find B. add C. isEmpty D. size E. contains

(b) (1**mark(s)**) A tree in which every node can have at most n children is referred to as a _____ tree. A. binary B. graph C. general D. n-ary E. ternary

(c) (1**mark(s)**) A full binary tree of height n has _____ leaves.
A. 2^n B. $3n$ C. $3(n+1)$ D. $\lg n$ E. $2(n+1)$

(d) (1**mark(s)**) Which of the following tree traversals traverses the subtrees from left to right and then visits the root? A. None of these B. Preorder C. Level-order D. Postorder E. Inorder

(e) (1**mark(s)**) Which of the following traversals is not easily implemented recursively? A. Inorder B. Level-order C. Postorder D. Preorder E. all of these are easily implemented recursively

(f) (1**mark(s)**) What property of the tree does its order specify?
A. maximum number of internal nodes B. maximum number of children per node
C. maximum number of leaves D. maximum number of edges E. maximum height

(g) (1**mark(s)**) Which of the following traversals never visits the root? A. Inorder
B. Postorder C. Level-order D. Preorder E. None of these

Question 4: Binary Search Trees ..... **6 points**

(a) (1**mark(s)**) In a binary search tree, the elements in the right subtree of the root are _____ the root element. A. less than B. equal to C. greater than D. less than or equal to E. greater than or equal to

(b) (1**mark(s)**) When removing an element from a binary search tree, we must always _____. A. remove all of its children B. An element should never be removed from a binary search tree. C. build a new tree D. make sure that the new tree is a binary search tree E. find its inorder successor

(c) (1**mark(s)**) When removing an element from a binary search tree that is a leaf, _____ will ensure that the resulting tree is still a binary search tree. A. replacing it with its only child B. replacing it with its inorder successor C. simply deleting it D. neither a, b, nor c E. all of the above

(d) (1**mark(s)**) When removing an element from a binary search tree that has a single child, _____ will ensure that the resulting tree is still a binary search tree. A. simply deleting it B. replacing it with its only child C. all of the above D. replacing it with its inorder successor E. neither a, b, nor c

(e) (1**mark(s)**) Adding an element to a binary search tree requires that the element must A. none of these B. have a positive value C. have a value that is in between the smallest and the largest value already in the binary search tree D. be an int E. be Comparable

(f) (1**mark**(s)) The balance factor of a node in an AVL tree is _____. A. the height of its right subtree minus the height of its left subtree B. the depth of the node from the root C. always positive D. the number of nodes below it E. the number of nodes above it

Question 5: Heaps and Priority Queues.....**5 points**

(a) (1**mark(s)**) When the removeMin operation removes the minimum element from a min-heap, that element is initially replaced by _____. A. the smaller of the left or the right child B. left child C. the element in the last node that was added D. none of these is correct E. right child

(b) (1**mark(s)**) Maintaining a heap as a complete binary tree means that A. the tree is degenerate B. None of these C. the tree is unbalanced D. leaf nodes are created at the lowest level in left- to right order E. every leaf node is at the same level

(c) (1**mark(s)**) A(n) _____ is a collection that is ordered using two ordering rules A. priority queue B. None of these C. interface D. exception E. stack

(d) (1**mark(s)**) A priority queue can be implemented using A. a list of queues or a min-heap or a maxheap B. a max heap C. a stack D. a list of queues E. a minheap

(e) (1**mark(s)**) In a maxheap, the largest element in the structure is always _____ A. an internal node B. a leaf C. the right child of the root D. the left child of the root E. the root

Question 6: Graphs ..... **5 points**

(a) (1**mark(s)**) A graph in which every edge is connected to every other edge is said to be _____. A. connected B. full C. sparse D. complete E. balanced

(b) (1**mark(s)**) A connected graph has which of the following properties?
A. For any pair of vertices, there is a path between them. B. Every vertex is adjacent to every other vertex. C. No vertex is adjacent to every other vertex. D. For any pair of vertices, there is an edge between them. E. There exists a vertex that is adjacent to every other vertex.

(c) (1**mark(s)**) A depth-first traversal of a graph uses which of the following data structures?
A. stack B. array C. queue D. None of these E. binary search tree

(d) (1**mark(s)**) A spanning tree of a graph is a tree that always has which of the following properties?
A. It includes all of the edges and some of the vertices of the graph. B. It includes some of the edges and all of the vertices of the graph. C. It includes some of the edges and some of the vertices of the graph. D. It includes all of the edges and all of the vertices of the graph. E. None of these

(e) (1**mark(s)**) A(n) _____ is a two-dimensional array that can be used to represent a graph.
A. graph node B. None of these C. adjacency list D. digraph list E. adjacency matrix

Part II :True & False Questions

Question 7: Recursion ..... **3 points**

(a) (1**mark(s)**) ___ All recursive methods must have a base case.

(b) (1**mark(s)**) ___ Recursive solutions to problems should be used whenever possible.

(c) (1**mark(s)**) ___ The Towers of Hanoi puzzle cannot be solved iteratively.

Question 8: Lists ..... **5 points**

(a) (1**mark(s)**) ___ A linked list is a conceptual notion of organising things in a linear manner.

(b) (1**mark(s)**) ___ An array is an example of an indexed list

(c) (1**mark(s)**) ___ The primary difference between an OrderedList ADT & UnorderedList ADT is in how elements are removed from the list

(d) (1**mark(s)**) ___ Using a circular array for an array-based implementation of a list would improve the performance of the operation to remove an element from the middle of a list

(e) (1**mark(s)**) ___ The operation to remove an element from an array implementation of a list collection is $\mathcal{O}(n)$.

Question 9: Binary Search Trees ..... **3 points**

(a) (1**mark(s)**) ___ An AVL tree is often implemented so that a node contains a reference to its parent node.

(b) (1**mark(s)**) ___ An ordered set of elements can be maintained using a linked list or a binary search tree. It is generally faster to locate an element in a binary search tree than it is to locate an element in a linked list.

(c) (1**mark(s)**) ___ A Red/Black tree is often implemented so that a node contains a reference to its parent node.

Question 10: Graphs ..... **3 points**

(a) (1**mark(s)**) ___ In order to create a topological ordering of vertices in a directed graph, the graph cannot have a cycle.

(b) (1**mark(s)**) ___ A graph is a special kind of tree.

(c) (1**mark(s)**) ___ A cycle is a path that starts and ends on the same vertex.

Section B

Question 11: Queues ..... **20 points**

(a) (5**mark(s)**) List the five basic operations on a queue and give the semantics of each operation.

(b) (2**mark(s)**) What is wrong with implementing a queue by using an array, where index 0 represents the front of the queue?

(c) (3**mark(s)**) Explain how a queue can be implemented using an array, where the *enqueue* and the *dequeue* operations are both constant time operations (for simplicity, assume that you will never need to expand the capacity of the array).

(d) (4**mark(s)**) Write an *enqueue* method for a queue implemented as a circular array. You may assume that you have access to a method called *expandCapacity* that will double the size of the array if necessary. The class has instance variables *front* and *rear*, which represent the indices of the *front* and *rear* of the queue. It also has an integer variable called *count* that represents the number of elements in the queue, as well as an array of generic *T* types called queue that represents the queue.

(e) (3**mark(s)**) The size of a queue using a linked list implementation is essentially unlimited. Is it possible to have an essentially unlimited size of a queue if an array-based implementation is used? Explain.

(f) (1**mark(s)**) Give an example of a real-life situation that can be modelled with a queue.

(g) (2**mark(s)**) What is gained by writing an interface to a QueueADT, as opposed to simply writing a Queue class, to solve a problem.

Question 12: Lists and General or Binary Search Trees ..... **20 points**

(a) (3**mark(s)**) What are the differences between an ordered list, an unordered list, and an indexed list?

(b) (2**mark(s)**) An academic researcher wants to use a Java class to create Student objects and arrange them in a list in descending order according to the value of their courseAverage member. What is the most appropriate list collection type to use?

(c) (3**mark(s)**) Explain how a level-order traversal of a tree is implemented.

(d) (2**mark(s)**) Explain how to add an element to a binary search tree.

(e) (3**mark(s)**) Draw a binary search tree that results from inserting the following elements: $\langle 12, 16, 9, 1, 15, 13 \rangle$

(f) (3**mark(s)**) Do the find and add operations on a binary search tree always require at most $\mathcal{O}(\lg n)$ comparisons? If so, why? If not, why not?

(g) (2**mark(s)**) In an AVL tree, each node keeps track of its own balance factor. What value(s) of the balance factor will trigger a re-balancing?

(h) (2**mark(s)**) If a node in an AVL tree requires re-balancing, what other nodes in the tree may also require re-balancing?

Question 13: Heaps, Priority Queues and Graphs ..... **20 points**

(a) (**2mark(s)**) What properties does a heap share with a binary tree?

(b) (**3mark(s)**) Explain how an element is added to a maxheap.

(c) (**3mark(s)**) Explain how heap sort works.

(d) (**4mark(s)**) What is the complexity of heap sort? How is it calculated?

(e) (**1mark(s)**) Where is the largest element in a minheap found?

(f) (**7mark(s)**) Use Prim's Algorithm to compute and draw the Minimum Spanning Tree for the undirected Graph $G = (V, E)$ where
$V = \{A, B, C, D, E, F, G\}$, $E = \{(A, B, 12), (A, D, 1), (A, F, 11),$
$(F, B, 6), (D, B, 9), (B, E, 4), (B, G, 3), (G, E, 7), (E, C, 2), (G, C, 5)\}$
and an edge is represented by using vertices and a non-negative weight.

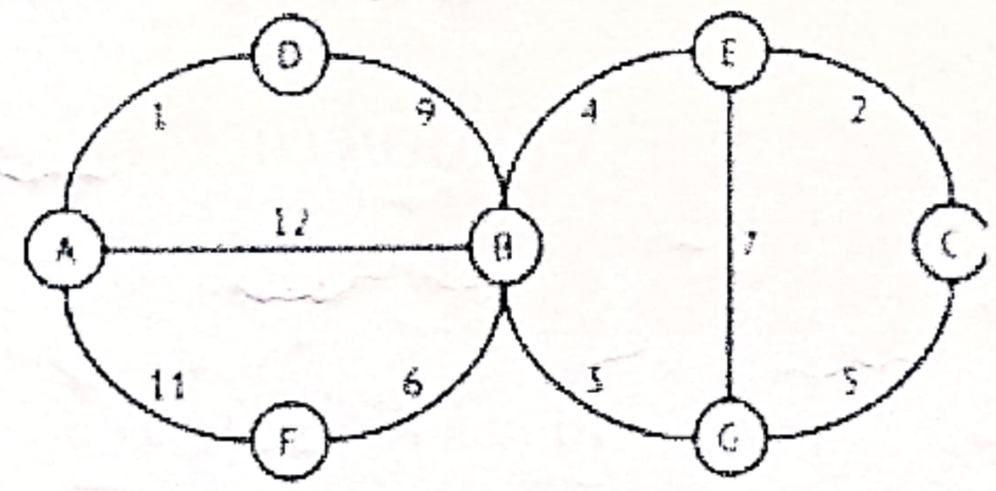

Figure 1: The Graph whose minimum spanning Tree you must compute

Note: You must show your working and next time there shall be no diagram for you!

Question 14: Problem Solving ..... **20 points**

(a) (**3mark(s)**) Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n, insertion sort runs in $8n^2$ steps, while merge sort runs in $64n \lg n$ steps. For which values of n does insertion sort beat merge sort?

(b) (**3mark(s)**) What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is $1000n$ on the same machine?

(c) Given three arrays X, Y & Z where each contains n integers (Note that integers can be positive or negative or zero).

i. (**4mark(s)**) Using pseudo code or Java or Python programming languages write an $\mathcal{O}(n^3)$ algorithm to find three integers a, b & c where $a \epsilon X$, $b \epsilon Y$ & $c \epsilon Z$ such that $a + b + c = 0$ (Hint: Brute force should do).

ii. (**5mark(s)**) Using BINARY-SEARCH (assume procedure is already implemented) improve the algorithm you developed in sub part i such that it runs in $\mathcal{O}(n^2 \lg n)$ time.

iii. (**5mark(s)**) Write an algorithm that solves the problem in $\mathcal{O}(n^2)$. (Hint: Clever sorting of any of the two arrays and a traversal strategy whose worst case running time is $\mathcal{O}(n)$ would do it.)