ROYAL UNIVERSITY OF BHUATAN
GYALPOZHING COLLEGE OF INFORMATION TECHNOLOGY
GYALPOZHING : BHUTAN

SEMESTER END EXAMINATION (AUTUMN 2020)

Class	:	**Bachelor of Science in Information Technology (Year II, Semester I)**
Module Title	:	**Algorithms and Data Structures**
Module Code	:	**ITS202**
Serial No.	:	**BSc(IT)/2020/III/F/ITS202/II**
Max. Marks	:	**50**
Max. Time	:	**3 Hours**

General Instructions:

1. Question paper has written component.
2. In no circumstances may you remove Answer Books, used or unused, from the Examination Room.
3. If the answer book is torn or folded or without Exam Cell's seal, report the matter to the Invigilator and get a new one.
4. Enter the required details such as Reg. Number, Module and other information as prescribed.
5. Do not write your name on any part of the Answer Book.
6. Number your answer according to the number assigned in the Question Paper.
7. Do not skip any pages when writing answers. Any rough sketches/calculations must be shown on the same page.
8. Do not fold or tear off any pages from the Answer Book. Any answer crossed by you will not be evaluated.
9. You may request for the supplementary Answer sheets only after the main answer Book is completely used.
10. A candidate who is found to have unauthorised materials in his /her possession, copying, talking or exchanging any material with others will be dealt with as per the Wheel of Academic Law.
11. No paper other than Admit Card will be allowed in the Examination Hall/Room unless otherwise specified in the Question Paper.

PART - I [10 Marks]

Answer all the questions

Multiple Choice Questions **[10 x 0.5 = 5]**

Q1) The Running time of Max_Heapify is

A. θ(nlogn) B. θ(n) C. O(n logn) D. O(logn) ✓

Q2) What is the speciality about the inorder traversal of a binary search tree?

A. It traverses in a non increasing order
B. It traverses in an increasing order ✓
C. It traverses in a random fashion
D. It traverses based on priority of the node

Q3) What is an AVL tree?

A. a tree which is balanced and is a height balanced tree ✓
B. a tree which is unbalanced and is a height balanced tree
C. a tree with three children
D. a tree with atmost 3 children

Q4) Which of the following is not a technique to avoid a collision?

A. Make the hash function appear random
B. Use the chaining method
C. Use uniform hashing
D. Increasing hash table size ✓

Q5) How many times must you tear a 1,024- page phonebook in half in order to whittle it down to a single page?

A. 8 B. 10 ✓ C. 32 D. 512

Q6) Sorting Algorithm which is the extension of insertion sort is

A. Selection Sort B. Shell Sort✓ C. Insertion Sort D. Bubble Sort

Q7) Graph Algorithm which doesn't work for Negative weights is

A. Bellan-Ford Algorithm
B. Dijkstra's Algorithm ✓
C. Kruskal Algorithm
D. Prims Algorithm

Q8) Which of the following operation take worst case linear time in the array implementation of stack?

A. Push
B. Pop
C. isEmpty
D. None ✓

Q9) The number of vertices in a Spanning tree with 12 edges is

A. 10
B. 11
C. 12
D. 13 ✓

Q10) One of the application of Stack data structure is

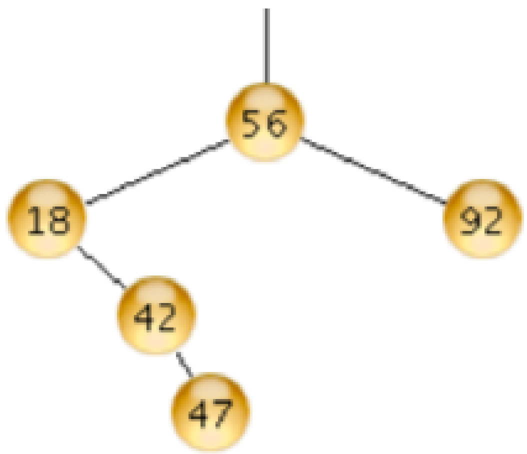
A. browser ✓
B. Apps
C. Studying
D. Staying in the line

Fill in the blanks **[5 x 0.5 = 2.5]**

Q11) A node can have any degree 0<= deg<= 2 and is called ....General Binary Tree........ **[0.5]**
Q12) A ....negative cycle... is a directed cycle whose sum of edge weights is negative. **[0.5]**
Q13) The time complexity of DFS is .....O(V + E)....... **[0.5]**
Q14) A .....connected component.... is a maximal set of connected vertices. **[0.5]**
Q15) A node has ...Data... and ...Link....... **[0.5]**

True or False **[5 x 0.5 = 2.5]**

Q16) top(), Returns the top element of the stack, without removing it (or null if the stack is empty). : True **[0.5]**
Q17) The time complexity of inserting a node in the head of the linked list is O(n). : False **[0.5]**
Q18) A queue is a first-in, first-out data structure. True **[0.5]**
Q19) Given Figure 1 tree is a AVL tree.: False **[0.5]**



Q20) Adding a constant to every edge weight does not change the solution to the single-source shortest-paths problem. False **[0.5]**

PART - II [15 Marks]

SHORT ANSWER QUESTION

Answer all the questions

Q1) For each algorithm below, specify an upper (O) and lower (ω) bound on its running time. Assume that the linked lists and arrays in question are all of length n. Do not assume that a data structure is sorted or unsorted unless told. **[5 * 0.5 = 2.5]**

For each row will be evaluated out of **[2* 0.25 = 0.5]**

Algorithms	**O**	ω
Sorting an array with Merge Sort	n log n	n log n
Sorting an array with Selection Sort	n^2	n^2
Inserting into a sorted Linked List	n	1
Searching a sorted array with Binary Search	log n	1
Searching a sorted Linked list with Linear Search	n	1

Q2) What is one property of a good hash function? **[1]**

Answer: Uniformly distributing some (possibly non-uniform) domain over a range.

Q3) What do you mean by Spanning tree of a Graph? **[1]**

Answer: A spanning tree of a grapg G is a subgraph T that is :

- Connected **[0.5]**
- Acyclic **[0.25]**
- Includes all the vertices **[0.25]**

Q4) Explain Edge relaxation with Example? **[2]**

Answer: Relax(u,v,w) **[1]**

- if d[v] >= d[u] + w(u,v)
- —> d[v] = d[u] + w(u,v)

Example **[1]**

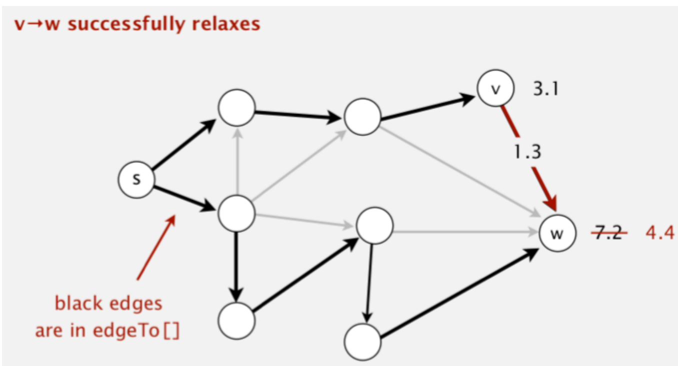


Figure 1: Example

Q5) Explain Hibbard Deletion of BST with the help of example. **[3]**

Answer: To delete a node with key k: search for node t containing key k.

- Case 0. [0 children] Delete t by setting parent link to null.

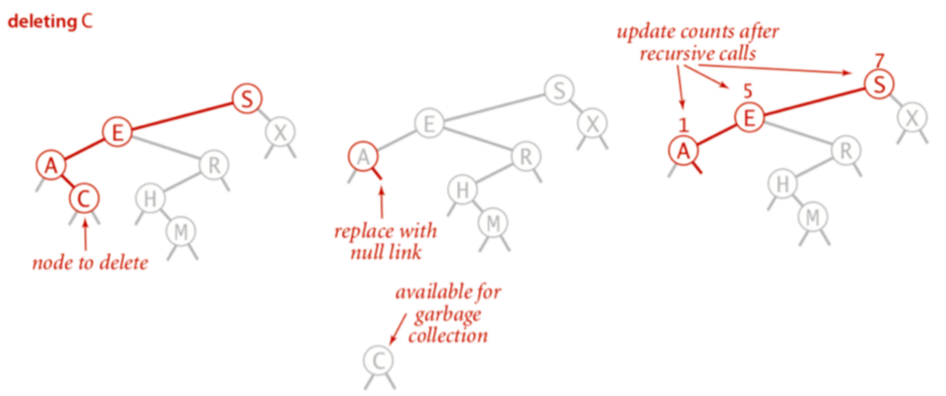


Figure 2: case 0

- Case 1. [1 child] Delete t by replacing parent link.

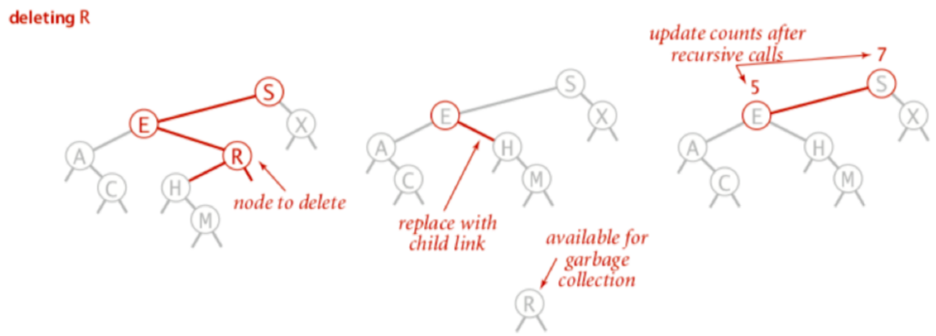


Figure 3: case 1

- Case 2. [2 children]
  - Find successor x of t. <— x has no left child
  - Delete the minimum in t's right subtree. <— but don't garbage collect x
  - Put x in t's spot. <—still a BST

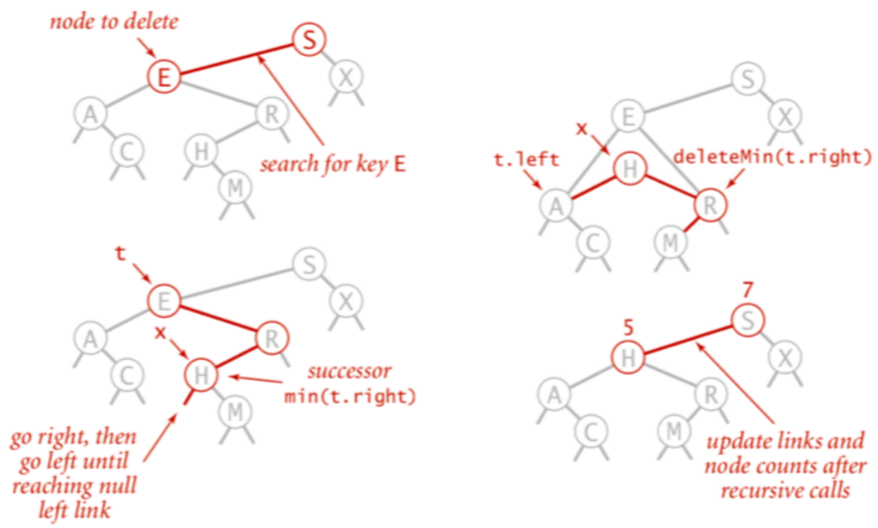


Figure 4: case 2

Q6) Explain Strictly Binary Tree with Example **[1]**

Answer: No node with degree one and each node should be with degree Zero or two. **[0.5]**

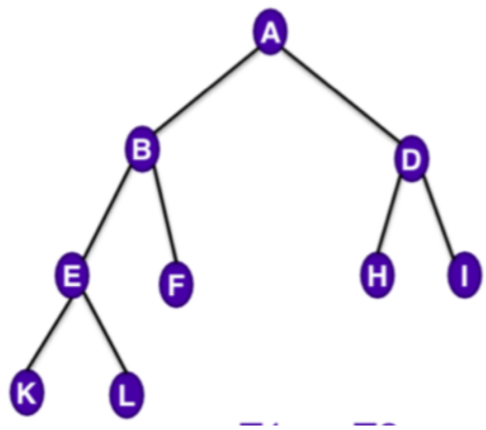


Figure 5: Strictly Binary Tree **[0.5]**

Q7) Consider the remarks below, each of which sounds like an advantage but is not without an underlying disadvantage too. Complete each of the remarks, making clear the price paid (i.e., tradeoff) for the advantage. **[1.5]**

i) Merge sort tends to be faster than bubble sort. Having said that ...
Answer: Having said that, merge sort requires twice as much space (to store values while merging). **[0.5]**

ii) A linked list can grow and shrink to fit as many elements as needed. Having said that ..
Answer: Having said that, a linked list does not allow for binary search, even if sorted, since it doesn't support direct access to nodes by index. **[0.5]**

iii) Binary search tends to be faster than linear search. Having said that ..
Answer: Having said that, binary search requires that its input be sorted, which might not be the case (and sorting it would require additional time). **[0.5]**

Q8) List down four steps involved in key-index counting. **[2]**
Answer: There are mainly four steps in processing key-indexed counting sort.

- Compute frequency counts. **[0.5]**
- Transform counts to indices. **[0.5]**
- Distribute the data. **[0.5]**
- Copy back. **[0.5]**

Q9) Mention the steps to insert data at the starting of a singly linked list? **[1]**
Answer: Steps to insert data at the starting of a singly linked list include:

- Create a new node
- Insert new node by allocating the head pointer to the new node next pointer
- Updating the head pointer to the point the new node

PART - III [25 Marks]

LONG ANSWER QUESTIONS

Answer all the questions

Q1) Apply Insertion sort on the given String **[5]**

E A S Y Q U E S T I O N

Answer:

		a[ ]											
i	j	0	1	2	3	4	5	6	7	8	9	10	11
		E	A	S	Y	Q	U	E	S	T	I	O	N
0	0	E	A	S	Y	Q	U	E	S	T	I	O	N
1	0	A	E	S	Y	Q	U	E	S	T	I	O	N
2	2	A	E	S	Y	Q	U	E	S	T	I	O	N
3	3	A	E	S	Y	Q	U	E	S	T	I	O	N
4	2	A	E	Q	S	Y	U	E	S	T	I	O	N
5	4	A	E	Q	S	U	Y	E	S	T	I	O	N
6	2	A	E	E	Q	S	U	Y	S	T	I	O	N
7	5	A	E	E	Q	S	S	U	Y	T	I	O	N
8	6	A	E	E	Q	S	S	T	U	Y	I	O	N
9	3	A	E	E	I	Q	S	S	T	U	Y	O	N
10	4	A	E	E	I	O	Q	S	S	T	U	Y	N
11	4	A	E	E	I	N	O	Q	S	S	T	U	Y
		A	E	E	I	N	O	Q	S	S	T	U	Y

Figure 6: Insertion sort

Q2) Apply Dijkstras Algorithm for the following Graph **[5]**

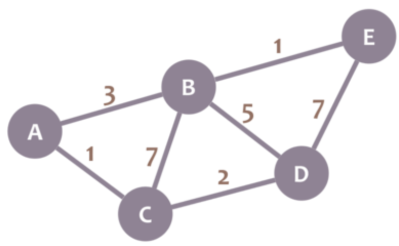


Figure 7: Dijkstras Algorithm Graph

Answer:

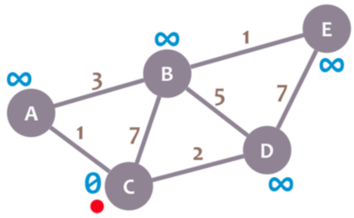


Figure 8:

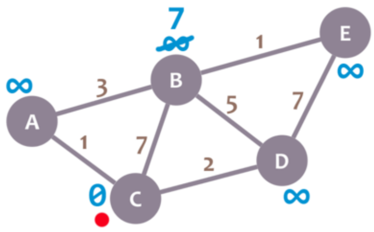


Figure 9:

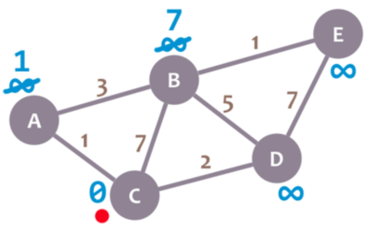


Figure 10:

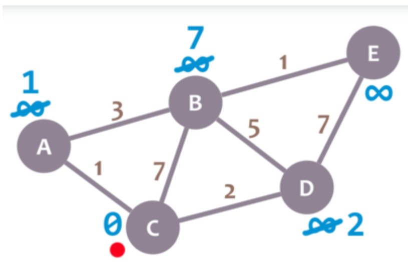


Figure 11:

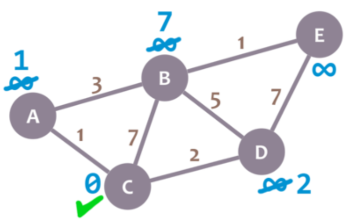


Figure 12:

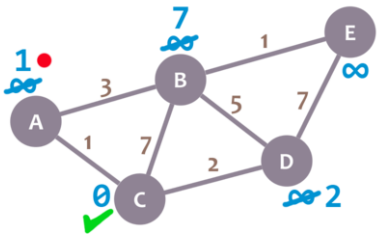


Figure 13:

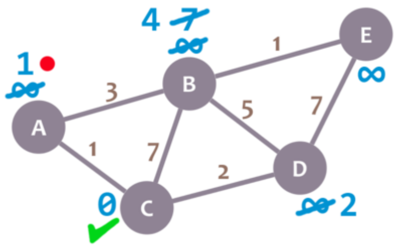


Figure 14:

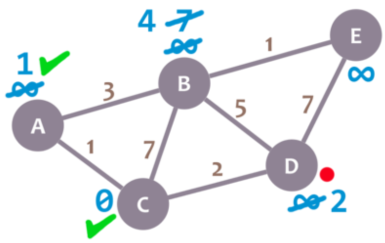


Figure 15:

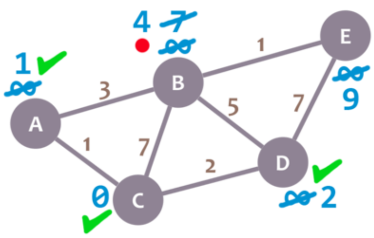


Figure 16:

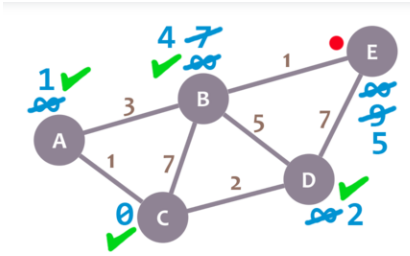


Figure 17:

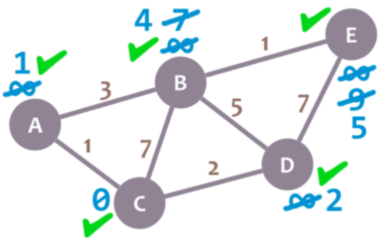


Figure 18:

Q3) Paro Airport runaway reservation system have only one runway available. Following details are provided for reservation.

- Reserve request: specifies landing time t.
- Add t to the set R of landing times if no other landings are scheduled within k minutes.
- k can vary: let's assume it is statically set (e.g. 3 min).
- After landing, remove request from R.
- What operations do we need in the data structure?
  - Adding requests. If they satisfy constraint!
  - Removing requests.
  - Notion of time, checks every m seconds to update the structure.
  - Nutshell: we need a data structure that allows for insertion and removal of elements.
  - Additional requirement: operations in O(lg n)

List and Explain with diagram all the data structure available comparing interms of their time complexity and finally suggest the best data structure to opt for given you are the developer of this system. **[5]**

Answer:

- Unsorted list/array: good? most operations are in O(n). Insertion can be in O(1) With diagram **[1]**

- Sorted list: Appending and sorting takes O(nlgn) time. Insertion takes O(n) time. A k minute check can be done in O(1) once the insertion point is found. With diagram **[1]**

- Sorted array: Binary search to find place to insert in O(lgn) time. Looks good? Insertion is still in O(n).We almost had it in the sorted list/array. With diagram **[1]**

- Key point: We need fast insertion into a sorted list.
- Binary Search Tree takes O(log n) time complexity. With diagram **[2]**

Q4) Perform Binary Search on this array to find the element 14. **[5]**

[1, 1,2,4,5,6,10,14,81,96,200]

Answer:

Target	Start	End	Middle
14	0	10	10/2 = 5

1	1	2	4	5	6	10	14	81	96	200
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

Figure 19:

Target	Start	End	Middle
14	6	10	(6+10)/2 = 8

1	1	2	4	5	6	10	14	81	96	200
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

Figure 20:

Target	Start	End	Middle
14	6	7	(6+7)/2 = 6

1	1	2	4	5	6	10	14	81	96	200
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

Figure 21:

Target	Start	End	Middle
14	7	7	(7+7)/2 = 7

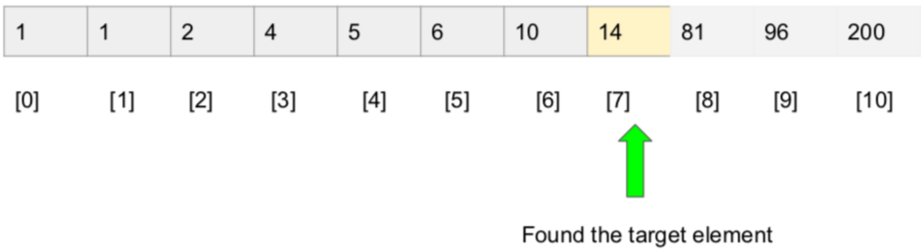


Figure 22:

Q5) List down the API of EdgeWeightedGraph with its Explanation. **[5]**

Answer:

`public class EdgeWeightedGraph`

	`EdgeWeightedGraph(int V)`	*create an empty graph with V vertices*
	`EdgeWeightedGraph(In in)`	*create a graph from input stream*
`void`	`addEdge(Edge e)`	*add weighted edge e to this graph*
`Iterable<Edge>`	`adj(int v)`	*edges incident to v*
`Iterable<Edge>`	`edges()`	*all edges in this graph*
`int`	`V()`	*number of vertices*
`int`	`E()`	*number of edges*
`String`	`toString()`	*string representation*

Figure 23: API