07

Library
Faculty of Technology
Rajarata University of Sri Lanka
Mihinthale

RAJARATA UNIVERSITY OF SRI LANKA
FACULTY OF APPLIED SCIENCES

B.Sc. (General) Degree in Information and Communication Technology

First Year Semester II Examination - Oct./Nov. 2017

ICT 1305 – DATASTRUCTURES

Time: Three (03) hours

Instructions:
Answer **All Five** Questions.
This paper consists of **three** (3) pages.

1. a. What are the factors that you should consider when selecting a data structure for an algorithm? (04 marks)

   b. Explain, using examples, how a selected data structure increases or decreases the efficiency of an algorithm. (04 marks)

   c. Explain the term *abstract data types* using an example. (03 marks)

   d. Suppose you need to read a given text and count the frequency of occurrences of each word appeared in the text. Suggest a suitable structure to store the list of words along with their frequencies. Explain its usage. (04 marks)

   [Total 15 marks]

2. a. Compare and contrast array based lists and dynamic lists. (02 marks)

   b. Suppose that a programmer has designed a dynamic list that only allows adding elements before the head element. Assume that a node of the list contains two integers, n1 and n2, and n1 is considered as the key.

      i. Write a function that removes the highest key element and inserts it into another list. (06 marks)

      ii. Write a recursive function to print the contents of the list in reverse order. (04 marks)

   c. Suppose that p and q are two consecutive elements of an array-based list. Write a few lines of pseudo code to insert a new element between p and q if the length of the array is n and the array is not full. (04 marks)

   d. Suppose that p is a middle node of a doubly linked list. Write a few lines of pseudo/C code to delete the node p. (04 marks)

   [Total 20 marks]

3. a. Briefly discuss how *queue* and *stack* data structures are used in computer Operating Systems to perform various operations. (04 marks)

   b. A programmer needs to read a set of student records and keep them in a queue until they are processed. A student record consists of index number and marks for five subjects. A processed record only contains the index number and the total marks of five subjects. Once a record is processed it is pushed to a stack.

   Write a few lines of C codes / C function for the followings:

   i. Implement required node structures of the *queue* and *stack* of the above scenario. (05 marks)

   ii. Implement a function to process a node of the queue and to return a stack node. (06 marks)

   c. If the programmer needs to print the set of processed records in descending order of their total marks, suggest a suitable method. (05 marks)

[Total 20 marks]

4. a. Compare and contrast *Queues* and *Priority Quests*. (03 marks)

   b. A *Priority Queue* can be implemented using a heap. Explain how to use heap property to implement a priority queue. (02 marks)

   c. Explain how the highest priority element is removed from an *array-based heap*. (04 marks)

   d. Illustrate how to insert the given list of integers [15, 12, 4, 20, 18, 2, 10 ,5] in to an array-based priority queue. Show the steps clearly. (05 marks)

   e. Explain the terms *in-place sorting* and *stable sorting*. (04 marks)

   f. Illustrate how to sort the given list of integers [15, 12, 4, 20, 18, 2, 10 ,5] using *merge sort* algorithm. (04 marks)

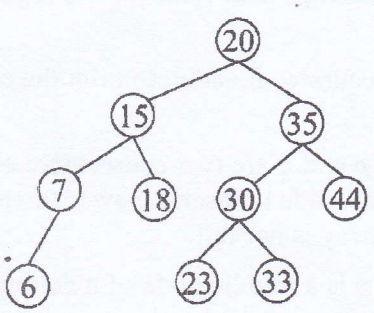
[Total 22 marks]

5. a. Explain how *binary search trees* (BST) differ from *trees*. (02 marks)

   b. If a BST node contains two fields, *height* and integer *key*, write a C-code to implement the structure of the BST node. (03 marks)

   c. Explain In-order, Pre-order and Post-order traversals of a BST using examples. (03 marks)

   d. Consider the given BST;



Apply the following algorithm to the given tree and write the expected output. Show the contents of the *queue*, *Q*, at each iteration. Assume that *enqueue* and *dequeue* functions are available. (08 marks)

```
Q=T->root //T - a BST, Q - A dynamic queue constructed
using BST nodes

While (Q<> NULL)

      Front=dequeue(Q)

      Print Front->key //key is the content of a BST node

      If(Front ->left<>NULL)

            enqueue (Q, Front ->left)

      If(Front ->left<>NULL)
```

e. What is an *AVL tree*? (02 marks)

f. Illustrate how to inset the list [15, 12, 4, 20, 18, 2, 10, 5, 7] in to an *AVL tree*. Show each step clearly. (05 marks)

[Total 23 marks]

-- End --