

BSc EXAMINATIONS 2003

DATA STRUCTURES

Wednesday 11 June, 2003

6.00 pm - 7.30 pm

Data Structures

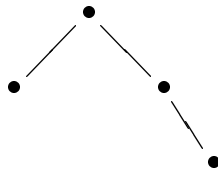
3 Questions

Candidates should answer ONLY TWO questions. The time allowed is ONE AND A HALF hours. Each question is worth 20 marks.

1. Give a brief explanation of each one of the following data structures. Describe the basic methods it should provide and how it can be implemented. Give an example of an application where each might be used.
 - (a) stack [4 marks]
 - (b) linked list [8 marks]
 - (c) hash table [8 marks]

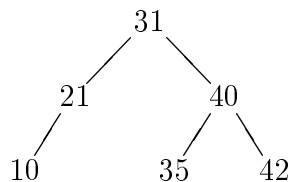
2. (a) Define the structure of a binary tree. [2 marks]
- (b) The number of nodes in a binary tree can be counted recursively by adding one to the sum of the number of nodes in its left and right descendant binary trees.
- (i) Write code to implement this `nrNodes` method. You may use Java or pseudo code. [2 marks]
- (ii) Define a **leaf** in a binary tree to be a node whose descendant binary trees are *both* empty. Write code to implement the `nrLeaves` method which counts the number of leaves in a binary tree. [3 marks]
- (iii) Write code for the `nrLayers` method which counts the number of **layers** in a binary tree. [3 marks]

For example, the binary tree



has 4 nodes, 2 leaves and 3 layers.

- (c) Define the structure of a binary **search** tree. [4 marks]
- (d) Describe a method for removing an element from a binary search tree, which ensures that the resulting tree is also a binary search tree. Show the effect of removing the root of the tree



using the method you have described.

[6 marks]

3. (a) Define the structure of a binary **heap**. [4 marks]
- (b) Describe an efficient algorithm to *insert* an element into a binary heap, ensuring that it remains a binary heap. [3 marks]
- (c) Describe an efficient algorithm to *remove* the top element of a non-empty binary heap, ensuring that it remains a binary heap. [3 marks]
- (d) Explain how an array of integers can be *sorted* by successively inserting its elements into an auxiliary binary heap. Show the successive states of the heap when sorting the array [31, 35, 40, 10, 42]. [4 marks]
- (e) Describe a way of sorting an array of integers, using heap operations, which requires no auxiliary storage. Indicate the successive states of the array when sorting [31, 35, 40, 10, 42]. [6 marks]