1.4 DATA TYPES & STRUCTURES · 1.4.2

Trees & traversals

Original practice questions · 30 marks · about 40 minutes · spec 1.4.2(b)(c)

Instructions. Answer all questions. Marks are shown in brackets [ ].

1

Total: 6 marks

(a) State what is meant by the *root* and a *leaf* node of a tree. **[2]**

(b) State what makes a tree a *binary* tree. **[1]**

(c) State the ordering rule that a binary search tree follows. **[2]**

(d) State one difference between a tree and a general graph. **[1]**

2

Total: 8 marks

The following numbers are inserted, in this order, into an empty binary search tree:
42, 23, 67, 12, 30, 55, 9, 35

(a) Draw the resulting binary search tree. **[3]**

(b) Write the **pre-order** traversal of your tree. **[2]**

(c) Write the **in-order** traversal of your tree. **[2]**

(d) State what the in-order traversal of a binary search tree always produces. **[1]**

3 **Total: 6 marks**

(a) Write the **post-order** traversal of the tree from Question 2. **[2]**

(b) Complete the recursive pseudocode below so that it performs an **in-order** traversal. **[4]**

```
procedure inOrder(node)
  if node.left ≠ null then _____________
  _____________
  if node.right ≠ null then _____________
endprocedure
```

4 **Total: 5 marks**

(a) Write the **breadth-first** (level-order) traversal of the tree from Question 2. **[2]**

(b) State the data structure typically used to implement a breadth-first traversal. **[1]**

(c) Explain the effect on search time of inserting values into a binary search tree in **sorted** order. **[2]**

5 **Total: 5 marks**

(a) Describe how the value **35** would be searched for in the tree from Question 2. **[3]**

(b) Explain one advantage of a (balanced) binary search tree over an unsorted list when searching for a value. **[2]**

END OF QUESTIONS