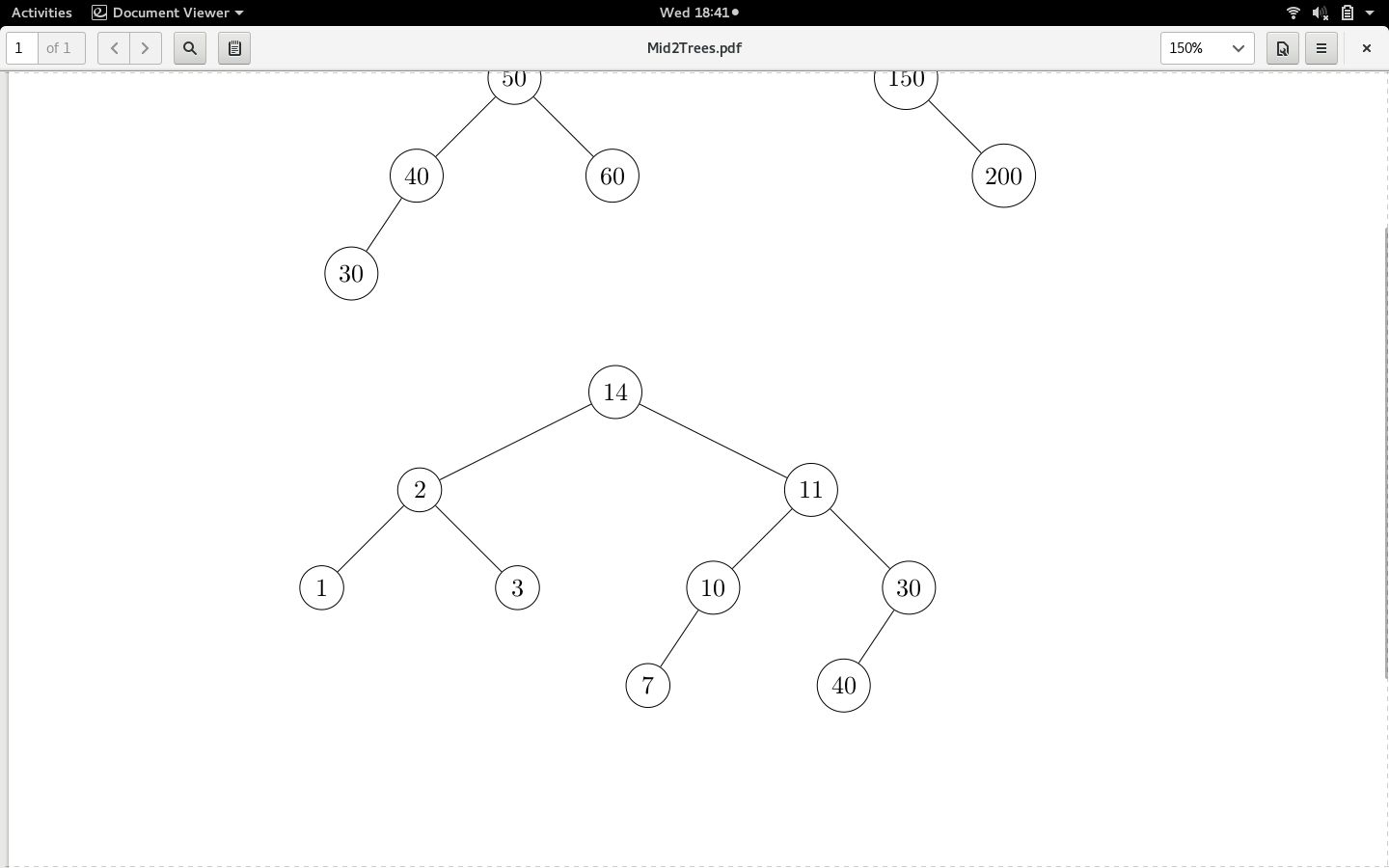
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | **Applied Programming** | **Course Code:** | **CS 319** |
| **Program:** | **MS (Computer Science)** | **Semester:** | **Fall 2017** |
| **Duration:** | **60 Minutes** | **Total Marks:** | **100** |
| **Paper Date:** | **02-Nov-17** | **Weight** | **15 %** |
| **Section:** | **N/A** | **Page(s):** | **4** |
| **Exam:** | **Midterm 2** | **Reg. No.** |  |
| **Instruction/Notes:** | * Please read the questions carefully before answering * Multiple Choice Questions (MCQs) **MUST** be marked on the question paper. All other questions must be answered in an answer script * You **MUST** return the question paper to the invigilator * No marks will be given for an MCQ, if there is overwriting * Write valid C++ code on programming problem(s) * Example tree given in section B, question 1 is just one example. You are responsible for any generalization that you make from it | | | |

**Section A - Multiple choice questions (Choose the most appropriate answer)**

1. The minimum number of nodes in a complete binary tree of height 3 is:
   1. 3
   2. 4
   3. 8
   4. 15
2. The maximum number of internal nodes in a complete binary tree of height 3 is:
   1. 3
   2. 5
   3. 7
   4. 15
3. The minimum height of a binary tree with 14 nodes is:
   1. 3
   2. 4
   3. 5
   4. 14
4. The height of the left and right subtrees in an AVL Tree may differ by:
   1. 0
   2. 1
   3. All of the above
   4. None of the above
5. A node in a complete binary tree is stored in an array at A[i]. Its right child will be stored at:
   1. A[i +1]
   2. A[i + 2]
   3. A[2 \* i]
   4. A[2 \* i + 1]
6. The pre-order traversal for the given binary tree visits nodes in the order:
   1. 1, 2, 3, 7, 10, 11, 14, 30, 40
   2. 1, 2, 3, 14, 7, 10, 11, 40, 30
   3. 1, 3, 2, 7, 10, 40, 30, 11, 14
   4. 14, 2, 1, 3, 11, 10, 7, 30, 40
7. The post-order traversal for the given binary tree visits nodes in the order:
   1. 1, 2, 3, 7, 10, 11, 14, 30, 40
   2. 1, 2, 3, 14, 7, 10, 11, 40, 30
   3. 1, 3, 2, 7, 10, 40, 30, 11, 14
   4. 14, 2, 1, 3, 11, 10, 7, 30, 40
8. The in-order traversal for the given binary tree visits nodes in the order:
   1. 1, 2, 3, 7, 10, 11, 14, 30, 40
   2. 1, 2, 3, 14, 7, 10, 11, 40, 30
   3. 1, 3, 2, 7, 10, 40, 30, 11, 14
   4. 14, 2, 1, 3, 11, 10, 7, 30, 40
9. Given a binary search tree, which traversal will print the nodes in sorted order:
   1. Pre order
   2. Post order
   3. In order
   4. None of the above
10. How many times is the symbol “#” printed by the call foo(4)?
    1. 3

void foo(int i){

if (i > 1){

foo( i / 2);

foo( i / 2);

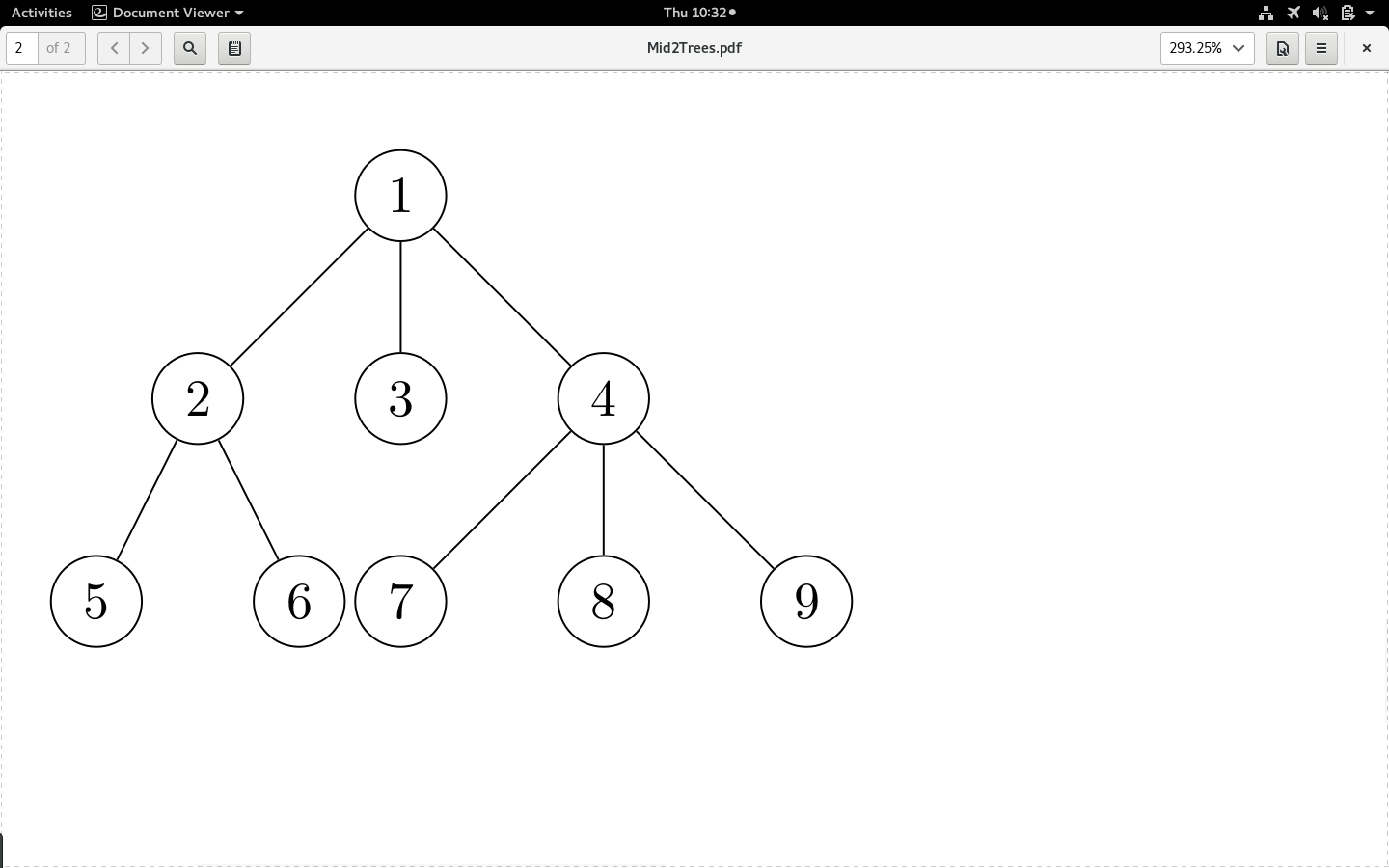
}

cout << “#”;

}

* 1. 4
  2. 7
  3. 8

**Section B**

Q. 1: An m-ary tree is a tree in which each node can have as many as m children. For a node v in an m-ary tree, define f(v) as the sum of that node’s key and its children’s keys. We need to find the max of f(v) over all v belonging to the m-ary tree. For the example tree shown below, the node with maximum f(v) is the node with key equal to 4 and your function should return 4 + 7 + 8 + 9 = 28. Write a recursive C++ function for this problem, assuming that the node structure is defined as:

struct Node{

int key;

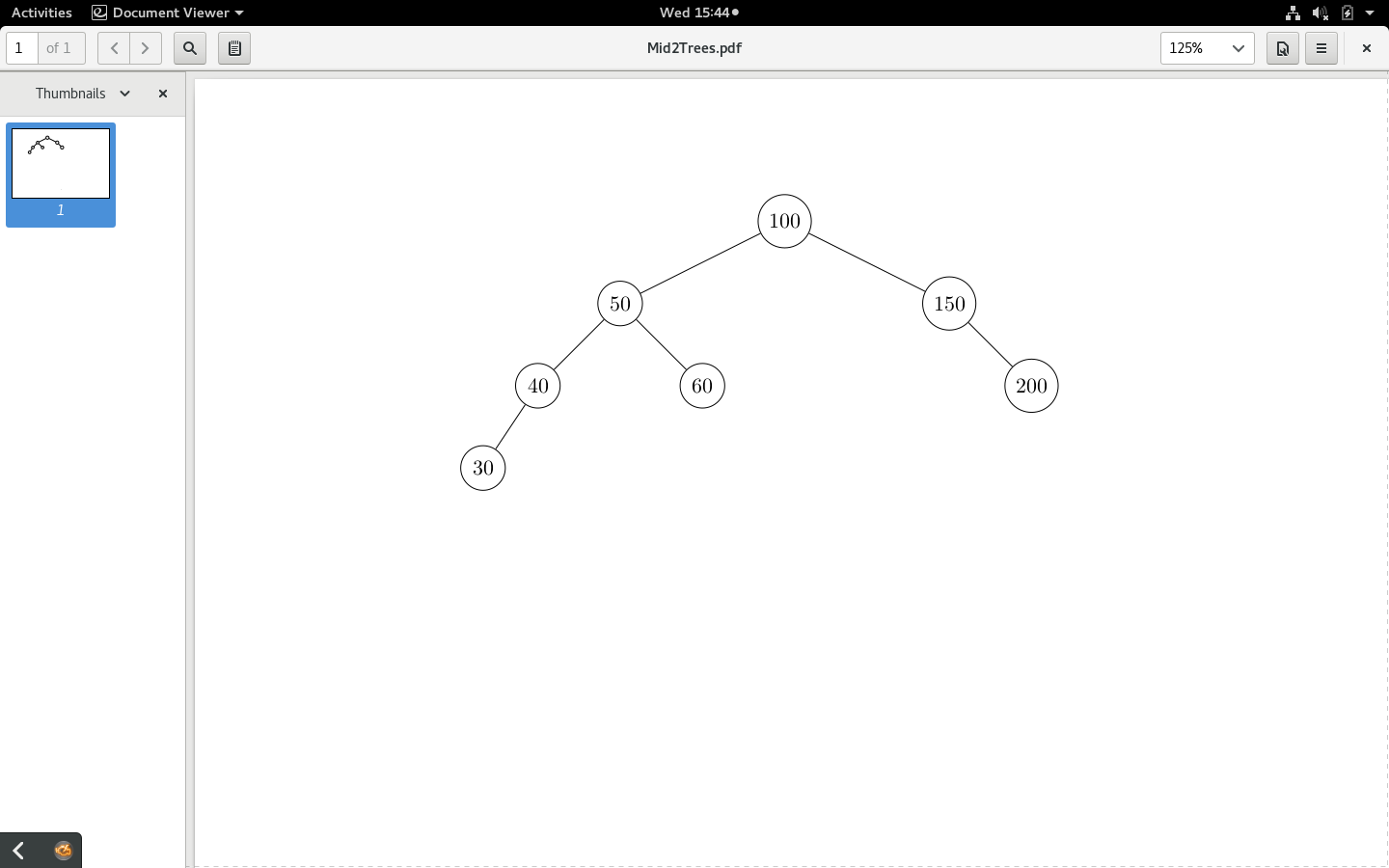
struct Node\* firstChild, \*nextSibling;

};

Q. 2: Solve the following recurrence using any appropriate method with the initial conditions a0 = 4 and a1 = 13:

an = 3 an-1 + 10 an-2

**Start the next three questions using the following AVL tree.**



Q. 3: Show the tree after the key 10 is inserted into it. Also show the intermediate steps.

Q. 5: Show the tree after the key 50 is deleted from it. Also show the intermediate steps.

Q. 6: Insert the keys 11, 9, 12, 14, 3, 15 and 7 (in that order) into an initially empty min heap.