**Homework 5 Solution**

|  |  |
| --- | --- |
| [Problem 1](http://www.cs.ucla.edu/classes/spring12/cs32/Homeworks/5/solution.html#P1) | [Problem 3](http://www.cs.ucla.edu/classes/spring12/cs32/Homeworks/5/solution.html#P3) |
| [Problem 2](http://www.cs.ucla.edu/classes/spring12/cs32/Homeworks/5/solution.html#P2) | [Problem 4](http://www.cs.ucla.edu/classes/spring12/cs32/Homeworks/5/solution.html#P4) |

**Problem 1:**

1. 50
2. 20 60
3. 10 40 70
4. 30 45 65 80
5. 25 35 75
6. One possibility is
7. 50
8. 25 60
9. 10 40 70
10. 35 45 65 80
11. 75

Other possibilities have the left subtree of 50 being

10

40

35 45

25

or

10

40

25 45

35

1. In-order: 10 20 25 30 35 40 45 50 60 65 70 75 80
2. Pre-order: 50 20 10 40 30 25 35 45 60 70 65 80 75
3. Post-order: 10 25 35 30 45 40 20 65 75 80 70 60 50

**Problem 2:**

1. 7
3. 3 5
4. 1 2 4
5. 7 3 5 1 2 4
6. 5 3 4 1 2

**Problem 3:**

1. struct Node
2. {
3. int data;
4. Node\* left;
5. Node\* right;
6. Node\* parent;
7. };
8. void insertAuxiliary(Node\*& n, int value, Node\* par)
9. {
10. if (n == NULL)
11. set n to point to a new Node whose data field is set to value,
12. whose left and right children are NULL, and whose parent field
13. is set to par.
14. else if (value < n->data)
15. insertAuxiliary(n->left, value, n);
16. else
17. insertAuxiliary(n->right, value, n);
18. }
19. void insert(Node\*& n, int value)
20. {
21. insertAuxiliary(n, value, NULL); // pass NULL as parent of root
22. }

**Problem 4:**

1. Adjacency matrix (entries shown as blank are 0; row is source, col is destination):
2. A B C D E F G H I
3. +-----------------
4. A | 1 1
5. B | 1
6. C | 1 1
7. D | 1 1
8. E | 1
9. F | 1 1
10. G | 1 1
11. H | 1
12. I | 1

Adjacency list:

A -> B C

B -> H

C -> D E

D -> B H

E -> G

F -> G I

G -> C H

H -> G

I -> C

B.

1. E G C D B H
2. E G C D H B
3. E G H C D B