Homework 5

* 1. 50  
      20 60  
      10 40 70  
      15 30 65 80  
      25 32 78
  2. Inorder: 10, 15, 20, 25, 30, 32, 40, 50, 60, 65, 70, 78, 70  
     Preorder: 50, 20, 10, 15, 40, 30, 25, 32, 60, 70, 65, 80, 78  
     Postorder: 15, 10, 25, 32, 30, 40, 20, 65, 78, 80, 70, 60, 50
  3. 50  
      20 60  
      10 40 70  
      15 25 65 80  
      32 78  
       
      50  
      15 60  
      10 40 70  
      25 65 80  
      32 78
  4. struct Node  
     {  
      int value;  
      Node\* parent;  
      Node\* left;  
      Node\* right;  
     };
  5. If tree is empty,  
      Add new node to root  
      Parent, left, and right are nullptr  
     Check root  
     While node has not been inserted yet,  
      If current value equals new node’s value,  
      Return  
      If current value is less than new node’s value,  
      If left pointer is not nullptr,  
      Check left pointer  
      Else,  
      Add new node to left pointer  
      Set parent to previous node  
      Set left and right pointers to nullptr  
      If current value is greater than new node’s value,  
      If right pointer is not nullptr,  
      Check right pointer  
      Else,  
      Add new node to right pointer  
      Set parent to previous node  
      Set left and right pointers to nullptr
  6. 7  
      5 6  
      4 0 2
  7. Array: [ 7 , 5 , 6 , 4 , 0 , 2 ]
  8. Array: [ 6 , 5 , 2 , 4 , 0 ]
  9. O(C + S)
  10. O(log(C) + S)
  11. O(log(C) + log(S))
  12. O(log(S))
  13. O(1)
  14. O(log(C) + S)
  15. O(C + Slog(S))
  16. O(C \* log(S))