Question 1

For this question I implemented the function using a for loop and an if statemet. The vector gave the parent of each value of the node. So if the parent is equal to a value in the vector, print the value.

The time complexity of this function is O(n) because it is solved linearly.

Question 2

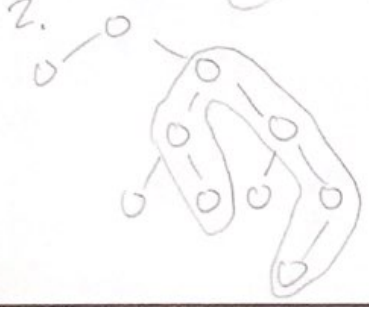
- Print tree is an in order traversal as it goes from left, Root, right.
- When pre order is called, it checks the tree from Root, Left, Right.
- when there are 100000 random nodes the max difference is 32

Question 3:

In order to slove this question there were two cases to look. Case 1 was if the root was in the diameter and case 2 was if the root was not in the diameter. For case 1, you use the previously implemented max depth to calculate the height of the left and right subtrees, then add 1 to find the diameter. For case 2, you find the diameter of the left and right subtree recursively to find the diameter. If the left diameter is greater, or right diameter is greater.

1

2.



Each node is visited recursively, so the time complexity is O(n) and space complexity is O(n)

Question 4

For this question I created a queueheader.h which included enqueue, dequeue, size, top, and display() functions. For

Question 5

First I checked if my tree was Null then I said if t->element was smaller than both x and y to call the function again with the left node. and vice versa if it was greater than both.

The time and space complexity of this function is O(n) because it visits each node recursively.

Makefile * queueheader.cpp is not part of the solution