1 Tree Traversals

Write the pre-order, in-order, post-order, and level-order traversals of the above binary search tree.

Pre-order: 
In-order: 
Post-order: 
Level-order (BFS): 

2 Tries

What strings are stored in the trie below? Now insert the strings indent, inches, and trie into the trie. Extra: How could you modify a trie so that you can efficiently determine the number of words with a specific prefix in the trie?

3 Heaps of Fun

(a) Assume that we have a binary min-heap (smallest value on top) data structure called Heap that stores integers, and has properly implemented insert and
removeMin methods. Draw the heap and its corresponding array representation after each of the operations below:

```java
Heap<Character> h = new Heap<>();
h.insert('f');
h.insert('h');
h.insert('d');
h.insert('b');
h.insert('c');
h.removeMin();
h.removeMin();
```

(b) Your friendly TA Anjali challenges you to quickly implement an integer max-heap data structure. However, you already have written a min-heap and you don’t feel like writing a whole second data structure. Can you use your min-heap to mimic the behavior of a max-heap?

*Hint:* Although you cannot alter them, you can still use methods from MinHeap.