Homework 3

1a. ALGORITHM computeRange (root)

// Input: root node

// output: range (difference between the largest & smallest numbers in the tree)

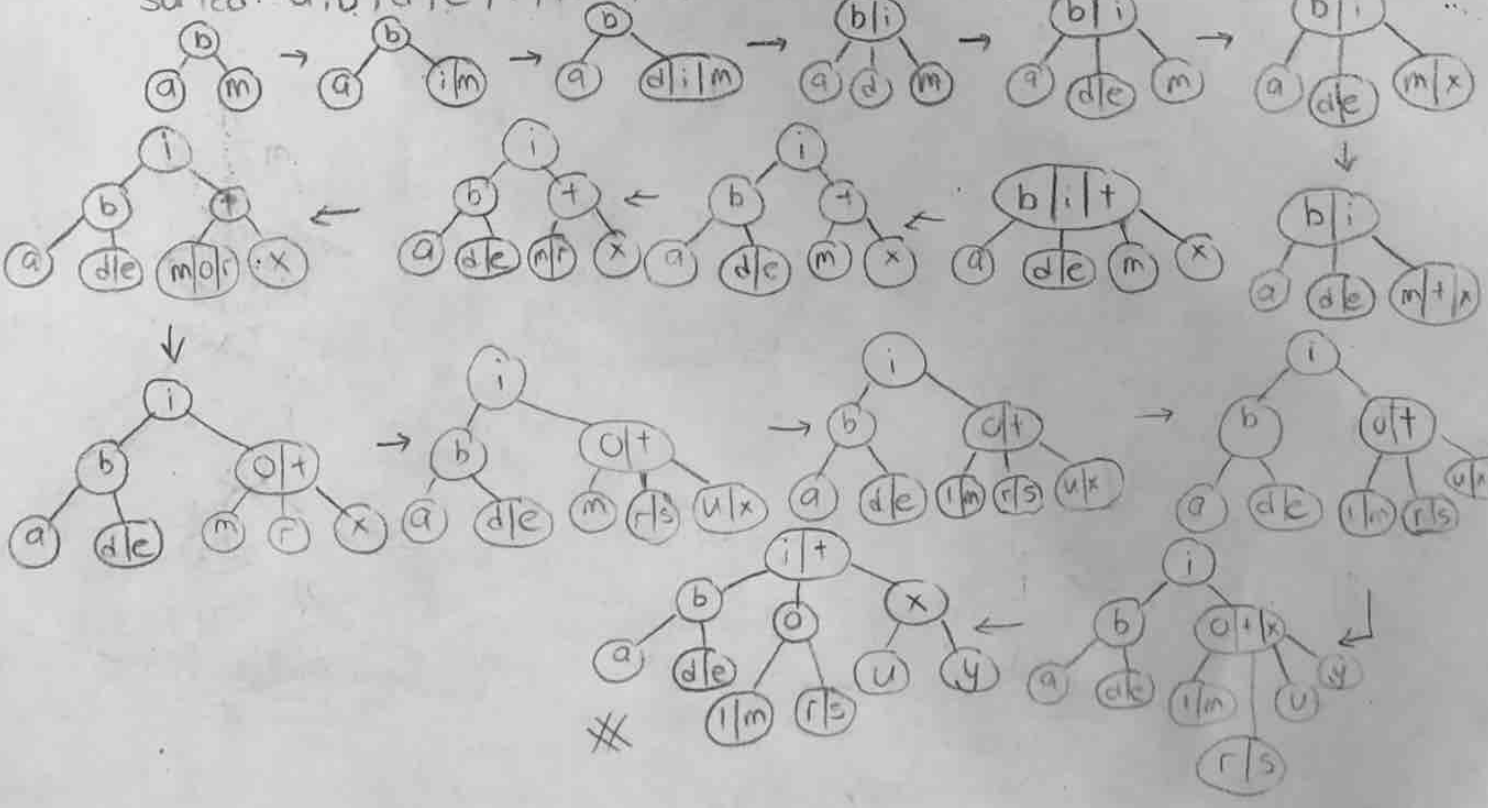
```
node ← root
while ( node.left ≠ null ) do
    node ← node.left
end while
smallest ← node.value
node ← root
while ( node.right ≠ null ) do
    node ← node.right
end while
largest ← node.value
range ← largest - smallest
return range
```

Worst case: $O(\log n) + O(\log n) \approx O(\log n)$

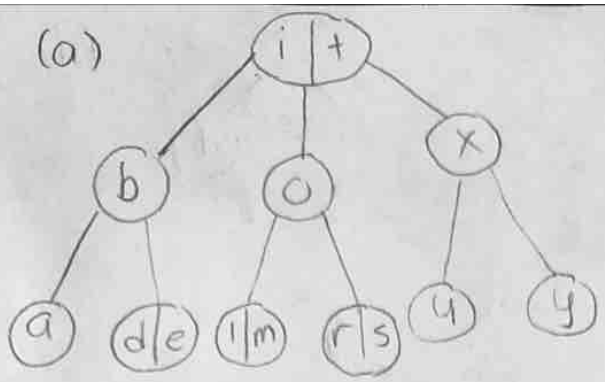
b. The smallest key is always located at the leftmost node, while the largest key is always located at the rightmost node.

2a. list; a, m, b, i, d, e, x, t, r, o, u, s, l, y

sorted: a, b, d, e, i, l, m, o, r, s, t, u, x, y



(a)



(b) - There are 14 keys in the AVL tree.

- The largest number of key comparisons in a successful search will be in the searches for e, m, and s will be equal to 5.

- Average number of comparisons:

$\frac{1}{14}C(i) + \frac{1}{14}C(t) + \frac{1}{14}C(b) + \frac{1}{14}C(o) + \frac{1}{14}C(x) + \frac{1}{14}C(a) + \frac{1}{14}C(d) + \frac{1}{14}C(e)$

$+ \frac{1}{14}C(l) + \frac{1}{14}C(m) + \frac{1}{14}C(r) + \frac{1}{14}C(s) + \frac{1}{14}(u) + \frac{1}{14}(y)$

$= (1 + 2 + 3 + 3 + 3 + 4 + 4 + 5 + 4 + 5 + 4 + 5 + 4 + 4)\ \frac{1}{14}$

$= \frac{51}{14}$

3 ALGORITHM computeRange (root)

// Input : root node

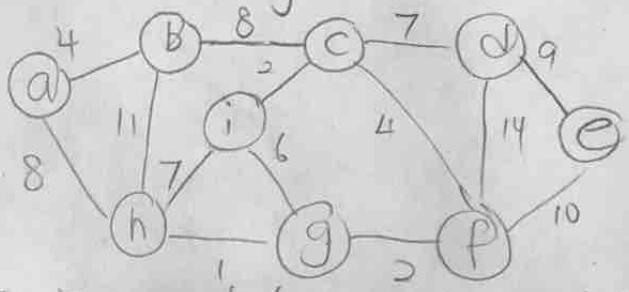
// Output : range (difference between the largest and smallest numbers in the tree)

```
node ← root
while (node.left ≠ null) do
    node ← node.left
end while
smallest ← node.value
node ← root
while (node.right ≠ null) do
    node ← node.right
end while
largest ← node.value
range ← largest - smallest
return range
```

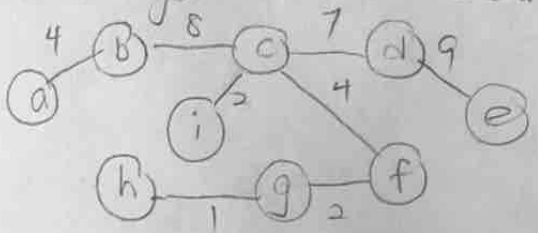
Worst case efficiency: $O(\log n) + O(\log n) \approx O(\log n)$

4. - No, it is not possible.

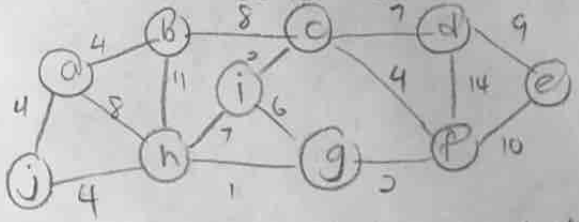
- If we assume that T is generated with vertices, $V_1, V_2, \ldots V_n$
- If a new edge is added and let V_{n+1} to be the vertex (V_1, V_{n+1})
- The diagram will change as (V_1, V_{n+1}) will replace (V_1, V_2) if the weight of (V_1, V_{n+1}) is smaller than (V_1, V_2)
- Therefore, the statement is wrong.
- set the diagram as



- T is generated as shown below



- If a new vertex and some edges are added at the beginning



- Tnew is generated as below which proves the statement is wrong