1. Analyze the \( \Theta \)-notation based time and space complexity of following program:

```plaintext
compute(k)
```

Where the procedure `compute` is defined as follows:

```plaintext
compute(m)
    if m \leq 1 then return
    x \leftarrow 0
    for i \leftarrow 1 to k do
        x \leftarrow x + 1
    print(x)
    compute(m/2)
```

2. Design and implement (in pseudocode) operation `list-merge(L1,L2)`
   - as input two linked lists \( L1 \) and \( L2 \)
   - within a list, elements are in ascending order
   - operation merges two lists to one, where all elements are in ascending order
   - after operation \( L1 \) becomes the merged list
   - after operation \( L2 \) is empty

   What is time/space complexity of operator?

3. (a) remove 38 from the tree in above figure
   (b) remove 18 from the tree in above figure
   (c) add firstly 43 and then 45 to the tree in above figure
   (d) remove 38 from the tree resulting in part (c)
   (e) remove 42 from the tree resulting in part (d)

   Draw tree after each insertion/deletion and show what fix-up-operation has the been used.