1. In each of the fragments of code below, how many times is the instruction \( \text{sum++} \) executed? In a, and b give an order of magnitude only, in c you need the exact number. You may use the formula \( \sum_{j=1}^{n} j = \frac{1}{2}n(n + 1) \)

a) \[ \text{for}(i = 1; i <= n; i++) \]
   \[ \text{for}(j = 1; j < i * i; j++) \text{ sum++;} \]

b) \[ \text{for}(i = 1; i <= n; i++) \]
   \[ \text{for}(j = 1; j <= i * i; j++) \text{ sum++;} \]

c) \[ \text{for}(i = 17; i <= 23; i++) \]
   \[ \text{for}(k = 0; k < 2 * i; k++) \text{ sum++;} \]

2. Give the rate of growth for each of the following functions. Your answer should be of the form \( \Theta(f(n)) \) for some fairly simple function \( f(n) \).

a) \( n^2 + (\ln(2^n))^3 \)

b) \( 5^n + 2^n + \sqrt{n} \) \hspace{1cm} // \( n + \sqrt{n} \) is in the exponent

c) \( n^3 + n^4 \left( \frac{\ln n}{\sqrt{n}} \right)^3 \)

d) \( n \sqrt{n} + 1^{0.8} + 2^{0.8} + 3^{0.8} + \ldots + n^{0.8} \)

e. \( (\ln(n!))^3 + (\sqrt{n})^{6.2} \)

3. The following sequence of numbers is inserted into an AVL tree. Show the shape of the tree after each insertion.

   \[ 10, 20, 30, 25, 26, 27 \]

4. Suppose you start with an empty AVL tree and insert a sequence of numbers. After each insertion, either no adjustments are made, or a rotation is performed. Give an example of a situation in which the first two successive rotations are \( RR \) and \( LR \). There maybe no rotations after some insertions.
5. Describe in some reasonable details how you remove a node from a binary search tree. Illustrate your explanation on the tree below: Show what happens after you remove the root. You should not write any code, just explain the “philosophy” of what happens.

![Binary Search Tree Diagram]

6. Suppose a binary tree contains an `int` as the data at each node. Write a method

```java
int nPositive(BNode r);
```

which will take as the parameter a reference `r` to a root of a binary tree, and return the number of nodes of the tree with root `r`, which contain a positive data.

7. Consider the iterative version of the merge sort, and consider the following array:

```
3, 18, 4, 45, 2, 6, 19, 5, 24, 6, 9
```

Show the state of the array after each insertion

8. Suppose you use the insertion sort to sort the following array:

```
2, 4, 6, 8, 10, 1
```

How many comparisons will be made? Exact answer please.

9. In a few sentences describe the advantages and disadvantages of the QuickSort.