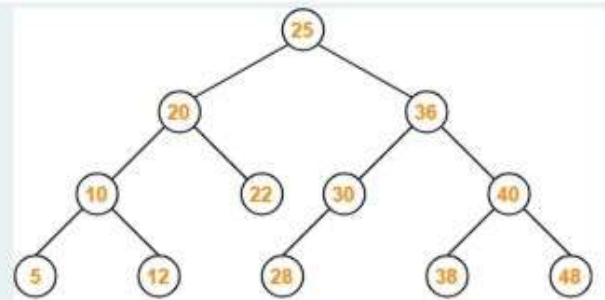


1

Write a simple function with tail recursion \*  
(0/10 Puan)

```
--
```

2



Show the trace (list the nodes visited) when searching for 34 in the binary search tree \*  
(0/10 Puan)

25-20-36-10-22-30-40-5-12-28-38-48

3

Show the results of the following sequence of events, by drawing the state of the data structure: add(2) add(5) add(1) add(7) add(8), add(8), remove(), remove()

Where add and remove are the operations that correspond to the basic operations in a stack \*  
(0/5 Puan)

251

4

What is the most important difference between the abstract class and an interface. \*  
(0/5 Puan)

if you use abstract class you can want method but interface must use interface method

5

In terms of Big-O complexity analysis what is the complexity of the term  $3 \log n + n + 5$  \*  
(0/5 Puan)

--

6

Write a recursive method for removing all the elements from a list \*  
(0/10 Puan)

---

7

For the arithmetic expression  $1 - 6 * 2 + 7 < (12 - (4 * 3)) + 8$  construct a binary tree and represent it in an array that can be used for the calculation of the result with the postorder traversal. Keep in mind operator  $\leq$  has lower precedence than  $+/-$  and  $*$  has precedence over  $+/-$ . Your answer is just the content of the array (Ex: -12\*678) \*  
(0/10 Puan)

--

8

For growable Array-based Array List implementation compare incremental strategy and the doubling strategy by analyzing the total time  $T(n)$  needed to perform a series of 20 push operations. Assume initial array size is 2. How many operations are required for each case. \*  
(0/10 Puan)

--

9

The number of operations executed by algorithms A and B is  $20 \cdot n \cdot \log n$  and  $2 \cdot n^3$ , respectively. Determine  $n_0$  such that A is better than B for  $n \geq n_0$ . \*

(0/10 Puan)

--

10

What is the complexity/growth rate of the following java function?

```
public static void printAll(double[] x , int a) {  
    int n = x.length;  
    for (int j=0; j < a; j++) {  
        for (int k=0; k < a; k++) {  
            System.out.print(x[j] + x[k]);  
        }  
    }  
} *
```

(0/5 Puan)

$n^2$

11

Show the results of the following sequence of events, by drawing the state of the data structure: add(2) add(5) add(1) add(7) add(8), add(8), remove(), remove()

Where add and remove are the operations that correspond to the basic operations in a queue

\*

(5/5 Puan)

1788

12

Write a recursive algorithm to compute the sum of all elements in an  $n \times n$  (two-dimensional) array of integers. What is your running time and space usage? \*

(0/15 Puan)

$n^2$