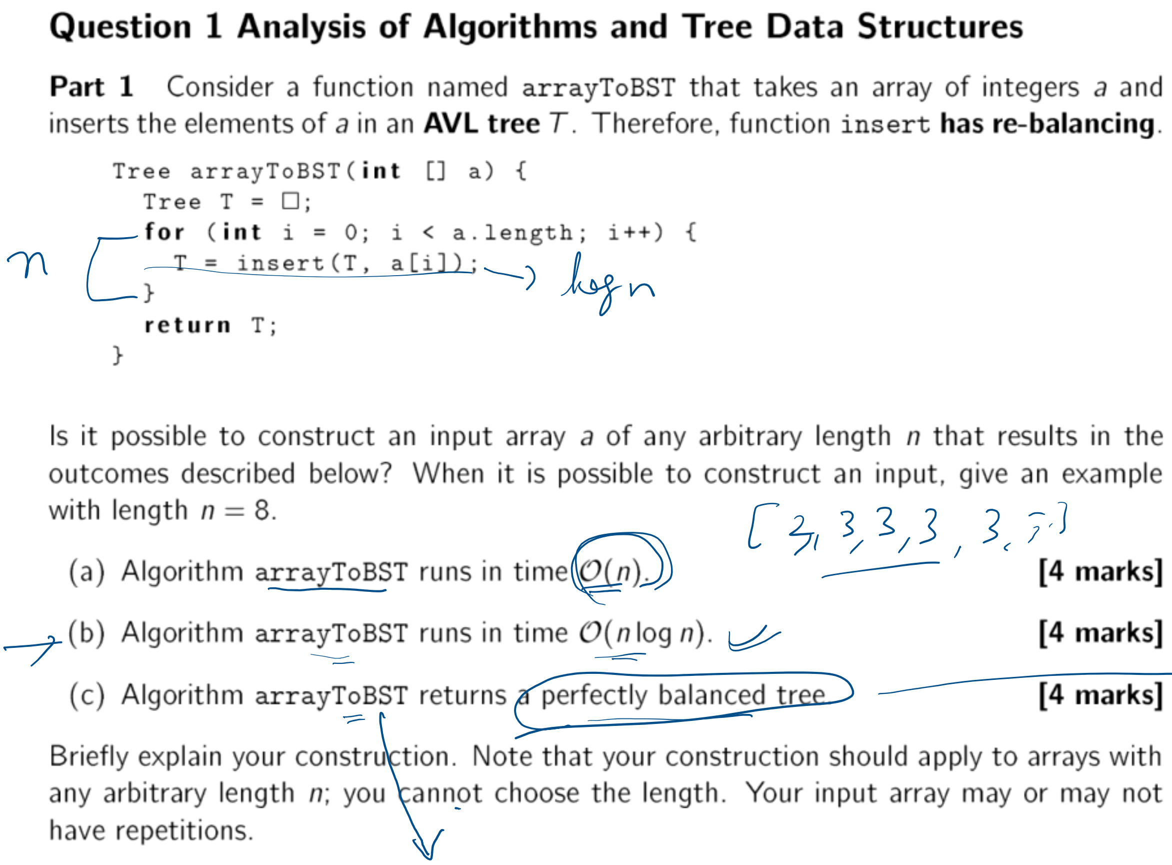
Q1 Part 1

Monday, 29 April 2024 7:31 PM



Question 1 Analysis of Algorithms and Tree Data Structures

Part 1 Consider a function named arrayToBST that takes an array of integers a and inserts the elements of a in an **AVL tree** T. Therefore, function insert **has re-balancing**.

```
Tree arrayToBST(int [] a) {
  Tree T = □;
  for (int i = 0; i < a.length; i++) {
    T = insert(T, a[i]);
  }
  return T;
}
```

n

→ log n

Is it possible to construct an input array a of any arbitrary length n that results in the outcomes described below? When it is possible to construct an input, give an example with length $n = 8$.

[3, 3, 3, 3, 3, ...]

(a) Algorithm arrayToBST runs in time $\mathcal{O}(n)$. **[4 marks]**

(b) Algorithm arrayToBST runs in time $\mathcal{O}(n \log n)$. ✓ **[4 marks]**

(c) Algorithm arrayToBST returns a perfectly balanced tree **[4 marks]**

No of Nodes $2^h - 1$ unique Elements

Briefly explain your construction. Note that your construction should apply to arrays with any arbitrary length n; you cannot choose the length. Your input array may or may not have repetitions.

it is impossible to construct a perfectly balanced tree with any arbitrary number of distinct elements, and repetition may be needed