Theoretical exercise III

Remarks: In all algorithm, always explain how and why they work. ALWAYS, analyze the complexity of your algorithms. In all algorithms, always try to get the fastest possible. A correct algorithm with slow running time may not get full credit. In all data structures, try to minimize as much as possible the running time of any operation.

1. **Questions 1:** Apply the Hoffman on the frequencies that are the 8 first Fibonacci numbers. are \( a : 2, b : 1, c : 2, d : 3, e : 5, g : 13, h : 21 \) and show the codes of all the letters.

2. **Question:** Say that the maximum frequency is at most twice the minimum frequency. Consider the order of magnitude of the sum of lengths of a Hoffman code. Is ot better than the order of magnitude of the sum of length of a ASCII code?

3. **Question 3:** Prove or disprove:
   (a) It is possible to get from any binary tree to any other binary tree using rotations?
   (b) It is possible to build a balanced (AVL) tree from a sorted array of \( n \) numbers in \( O(n) \)?
   (c) Is it possible to build a binary search tree from a non sorted array in time \( O(n) \)?
   (d) It is possible to build a balanced AVL tree from any given binary search tree, in \( O(n) \) time?

4. **Question 4:** Give data structures for the following:
   (a) **Insert** \((S, x)\): insert \( x \) to \( S \)
   (b) **Search** \((x)\): search if \( x \) is in \( S \)
   (c) **Delete** \((S, x)\): Delete \( x \) from \( S \).
   (d) **Max − Gap** \((S)\): find the maximum difference, \( \max_{x,y \in S}\{|x − y|\} \), in \( O(1) \) time.
   (e) **Min − Gap** \((S)\): find in \( O(1) \) time, \( \min_{x,y \in S, x \neq y}\{|x − y|\} \).

5. **Question 5:** Given two arrays of size \( n \) with pairwise distinct elements have the same elements (perhaps in a different order). if the entries in \( A \). Remark: The next topic will allow us a faster algorithm.