Problem Set 2

1. Let A be an array of n elements with the following property: there exists an $i \in [0, n-1]$ such that $A[0] < A[1] < \ldots A[i-1] < A[i] > A[i+1] > \ldots A[n]$. Show how to find such an i in $O(\log n)$ time.

2. Given a balanced binary search tree on n nodes and a target sum, write a function that returns true if there is a pair with sum equals to target sum, otherwise return false. Expected time complexity is $O(n)$ and only $O(\log n)$ extra space can be used. Any modification to binary search tree is not allowed.

3. You are given two binary search trees (not necessarily balanced). Design an algorithm that merges the two given trees into a balanced binary search tree in linear time.

4. Prove that the result of inserting any increasing sequence of $2^k - 1$ numbers into an initially empty AVL tree results in a perfectly balanced tree of height $k - 1$.

5. Call a family of trees balanced if every tree in the family has height $O(\log n)$, where n is the number of nodes in the tree. For each property below, determine whether the family of binary trees satisfying that property is balanced. If you answer is "no", provide a counterexample. If your answer is "yes", give a proof.

   - Every node of the tree is either a leaf or it has two children.
   - The size of each subtree can be written as $2^k - 1$, where k is an integer (k is not the same for each subtree).
   - There is a constant $c > 0$ such that, for each node of the tree, the size of the smaller child subtree of this node is at least c times the size of the larger child subtree.
   - There is a constant c such that, for each node of the tree, the heights of its children subtrees differ by at most c.
   - The average depth of a node is $O(\log n)$. (Recall that the depth of a node x is the number of edges along the path from the root of the tree to x.)

6. Given an array of n elements, where each element is at most k away from its target position (its position in sorted array), design an algorithm that sorts the array in $O(n \log k)$ time.