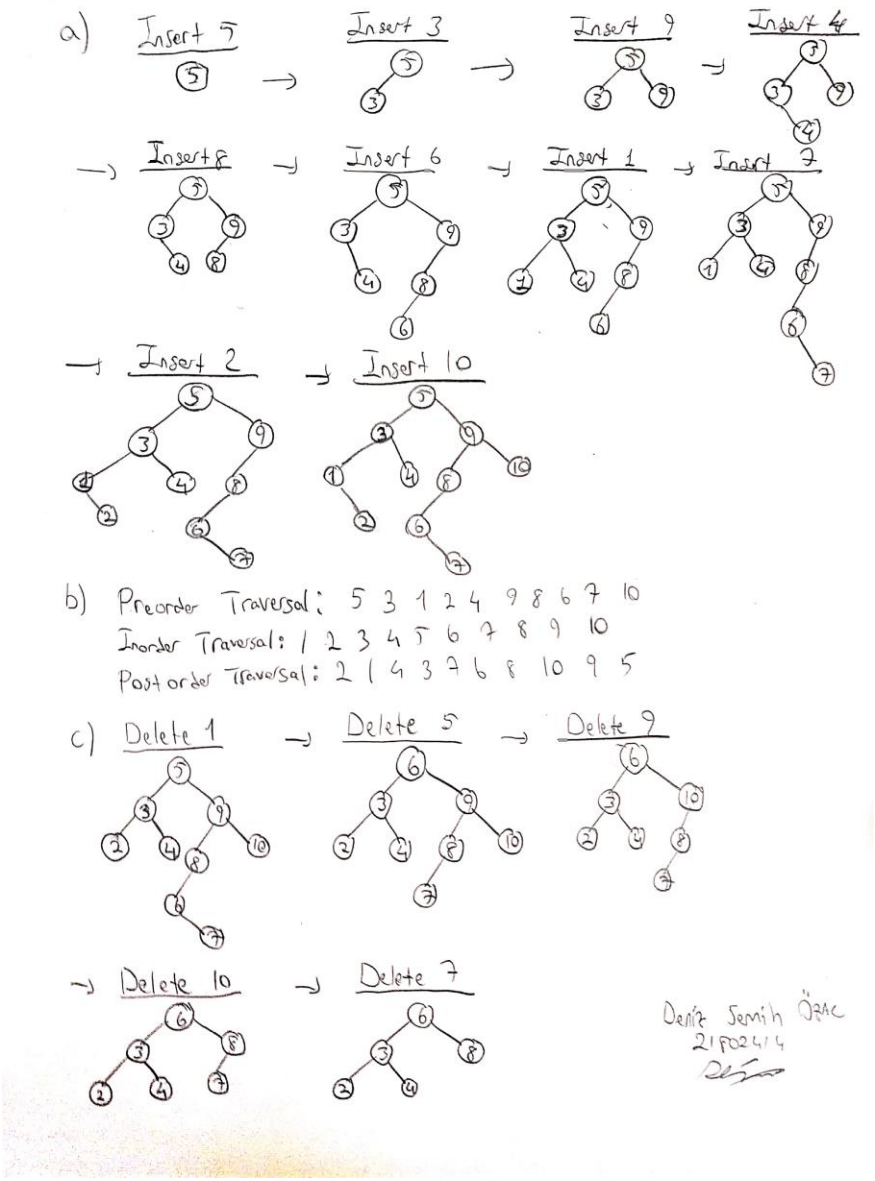
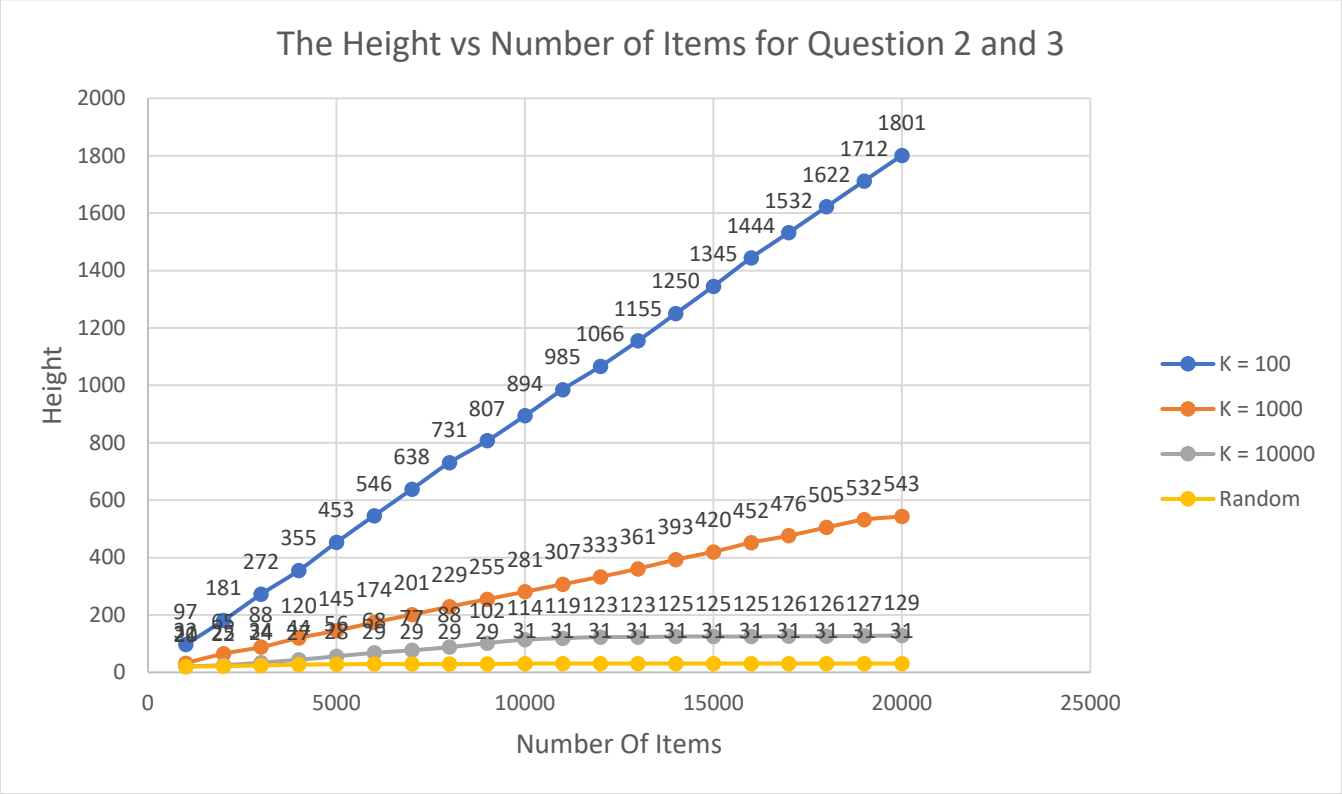
Title: Binary Search Trees
Author: Deniz Semih Özal
ID: 21802414
Section: 1
Assignment: 2
Description: Height Analysis in Binary Trees

Question 1)



Question 3)

a) The Height of the Tree vs Number of Items of the Tree Graph for Question 2 & 3



Conclusion:

As it could be seen from the graph, random sorted array has quite efficient in terms of height and time complexity. Theoretically, since random sorted array has a greater possibility that having less or greater nodes than its parents, binary search tree is quite efficient for random arrays. Similarly, when the K is 10000 nearly sorted array will be a kind of random sorted array and again its result is quite efficient. Time complexity is expected about O(n) for the worst case and O(logn) for best-average cases. It can be clearly seen that random sorted array and nearly sorted array with K = 10000 is quite close to the O(logn) and their height is very low. On the other hand, K = 100 is th worst case since array is a ascending (or descending) sorted array nevertheless the result is close to the average case. However, when K is approaching to the zero, the height would increase dramatically and even my compiler cannot handle it and gave me an error. Therefore, worst case could be much more worse compared to the above graph. Consequently, binary search is very efficient data structure for random sorted arrays but it is not quite efficient for ascending or descending sorted arrays.