

1. Let T be a proper binary tree with height h and n nodes. Prove that $\log(n+1) - 1 \leq h \leq (n-1)/2$.
 2. Give a recursive and a non-recursive algorithm for binary search.
 3. Consider the following sequence of keys: (5, 16, 22, 45, 2, 10, 18, 30, 50, 12, 1). Illustrate with the help of a diagram the insertion of these keys in the given order into:
 - An initially empty (2,4) tree.
 - An initially empty red-black tree.
 4. Develop an algorithm that computes the k^{th} smallest element of a set of n distinct integers in $O(n + k \log n)$ time.
 5. Is it true that if the height of an AVL tree is increased by two, then the minimum possible number of stored keys atleast doubles. If you agree with this then prove it, otherwise give a counter example.
 6. Prove that the height of an AVL tree storing n items is $O(\log n)$.
 7. Prove that a multiway search tree storing n items has $n + 1$ external nodes.
 8. Prove that the height of a 2-4 tree storing n items is $\Theta(\log n)$.
 9. Let T and U be (2-4) trees storing n and m items respectively, such that all the items in T have keys less than the keys of all the items in
-

U . Describe an $O(\log n + \log m)$ time method for joining T and U into a single tree that stores all the items in T and U (destroying the old versions of T and U).

10. Prove that the height of a red-black tree is $\Theta(\log(n + 1))$.