60

47

50 | 51

37 | 40

33

49 | 55

4 | 26

35

30 | 45

28

8 | 11 | 19

3

**2-)**

a-)

max number of nodes can be: (h3 -1)

b-)

when “i” inserted tree will be expand.

g | h

b | d | f

e

c

a

a

c

d

b

f

e

g | h | i

c-)

It is same running time with binary search tree sorting which is O(n logn) because they have same order. Smaller value in the left child, grater value in the right child.

d-)

Since root have to be black node and if root has a red child then we can easily say, not each subtree is a red-black tree. Eg. 30 and 26 are in same node(first tree in that document).

**3-)**

1. First, I will create new array with n size then I will fill that array with separate chaining (hashing). After array filled, that will take O(n) time (O(1) time for each item), I will search the hash table. For e

Linear probing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Content | 30 | 15 | 22 | 11 | 14 | 18 |  |

Quadratic probing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Content | 30 | 15 | 22 | 11 | 14 | 18 |  |