

Homework 1 Solution

Q1.

4 ,If we search for number 34, we will first compare it with the number 5 in the middle, then with the number 13 in the middle of the right half, then with the number 21, and then with the number 34. This is because when we have even number of elements we compare with the left of the two middle elements. If we compared with the right of the two middle elements, then to find number 1 we would compare it first with number 8, then with number 3, then with number 2 and then with number 1 - also 4 comparisons.

Q2.

(10,3,15,12,7,9,6,12)
(10,3,15,12,)(7,9,6,12)
(10,3)(15,12)(7,9)(6,12)
(10)(3)(15)(12)(7)(9)(6)(12)
(10,3)(15,12)(9,7)(12,6)
(15,12,10,3)(12,9,7,6)
(15,12,10,12,9,7,6,3)

Merge Sort visualize:

<https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/visualize/>

Q3.

<https://www.youtube.com/watch?v=UfC9BfVgrOE>

Q4.

-Algorithm

- Sort the array using quick sort and create a variable count and previous , $prev = INT_MIN$.
- Traverse the element from start to end.
- If the current element is equal to the previous element increase the count.
- Else set the count to 1.
- If the count is greater than half the size of array, print the element as majority element and break.
- If no majority element found, print "No majority element"

<https://www.geeksforgeeks.org/majority-element/> <- method 5 (quick sort)

<https://www.geeksforgeeks.org/quick-sort/> (quick sort)

Homework 1 Solution

Q5.

Inorder traversal of the constructed tree:

H J E L M Q V I S T G R Y B Z O C K P F U N D ß W A X

[Construct Full Binary Tree from given preorder and postorder traversals](#)