Assignment

B.S. Udaykiran
(1923111185)

(1) Given an array of {4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6 -8, -9} integers. find the maximum and minimum product that can be obtained by multiplying two integers from that array.

Sol:- array is {4, -2, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1}

we need to consider the largest and smallest products that can be formed by selecting numbers from array.

2. Sort the array
3. Identify possible candidates for maximum product.
4. Identify possible candidates for minimum product

Calculating maximum product:-

* The two largest positive numbers are 10 & 11 $\Rightarrow$ $10 \times 11 = 110$
* The two smallest negative numbers are -9 and -8.

$-9 \times 8 = 72$.

The maximum product is 110.

The largest positive and negative number is 11 or -9

$11 \times -4 = -44$

The smallest negative numbers are

$-9 \times -8 = 72$

-99 is smaller than 72 so

maximum product = 110

minimum product = -99

(2) Demonstrate the binary search method to search the key = 33 from array = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91}

Sol:- Given key = 33 and array = {2, 5, 8, 2, 16, 23, 38, 56, 72, 91}

1. initialize points

low = 0 and high = 9

Calculate mid $= \left[\frac{low + high}{2}\right] \cdot \left[\frac{6+4}{2}\right] = 5$

Compare arr [mid] with key:

arr [n] = 16

Since 16 < 23 update low = mid + 1 = 5

Calculate mid $= \left(\frac{low + high}{2}\right) = \frac{5+9}{2} = 7$

Compare arr [mid] with key.

arr [7] = 56

Since 56 > 23 update high = mid − 1 = 6

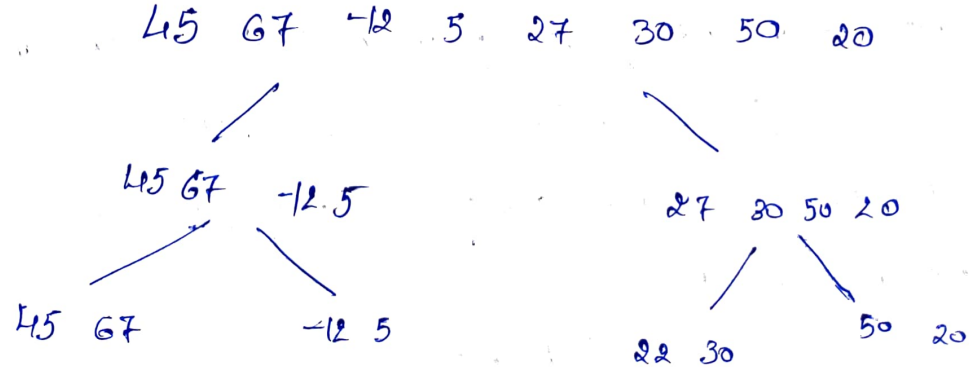
arr [mid] = arr [5] = 23

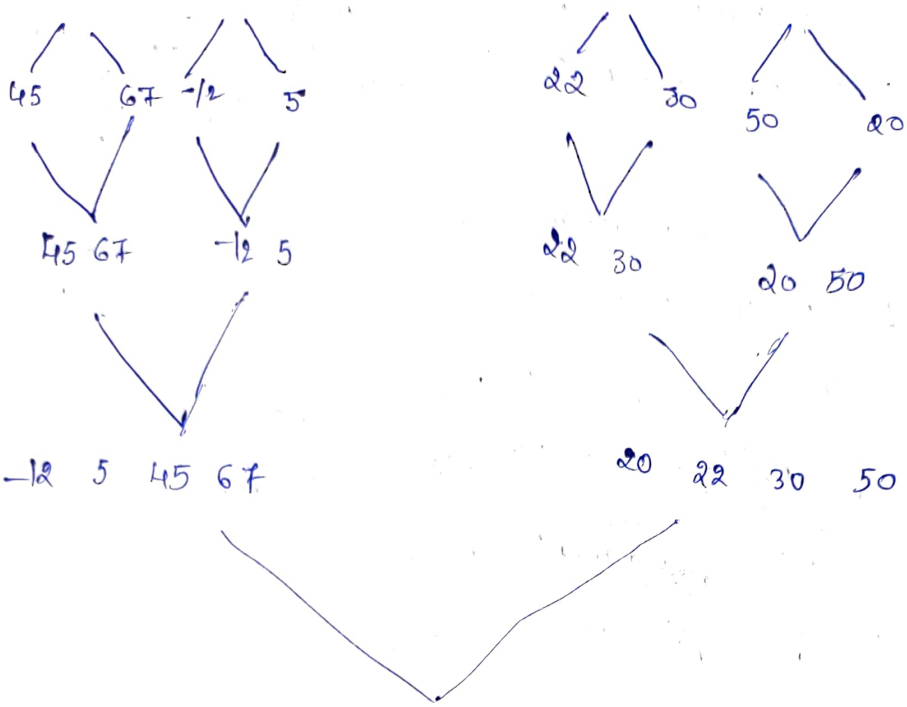
23 == 23 The key is found at index 5.

∴ The key = 23 is found at index 5

(3) Apply merge sort and other list of 8 elements, set up the recurrence relation to the number of key comparisons made by merge sort.

Sol:- Merge Sort:-





Sorted list :- (-12, 5, 20, 23, 30, 45, 50, 67)

(4) Find the no. of times to perform for selection sort.

Sol:- Recurrence relation to comparisons

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

if $n=1, T(1)=0$ Base case

At each level of recursion we makat most $n-1$ comparison to merge two lists of size $n/2$ so it comes $t(n) = 2t\left(\frac{n}{2}\right) + n-1$

Solving recurrece relation we get

$$T(n) = n\log_2(n) - n + 1$$

The recurrence relation is $T(n) = 2T(n/2) + O(n)$ or more precisely $T(n) = n\log_2(n) - n + 1$

(5) Find the no of times to perform solving swapping for selection sort also estimate the time complexity for order.

Sol:- The selection sort algorithm always makes exactly n-1 swaps in worst case, where

n is the no. of elements in list

givens = {12, 7, 5, -2, 18, 6, 13, 4}

no. of elements, n = 8

no. of swaps = n-1 = 8-1 = 7

Time complexity :- The time complexity of selection sort in Big O not ration is $O(n^2)$

so, the number of swaps is 1, and the time complexity is $O(n^2)$