ASSIGNMENT-12

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

Solution:

Array is [4,-2,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9]

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

1. Sort the array

Sorted array

[-9,-8,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11]

2. Identify possible candidates for maximum product
3. Identify possible candidates for minimum product.

Calculating maximum products

- The two longest positive numbers are 10 and 11

  10x11=110

- The two smallest negative numbers are -9 and -8

  -9x-8=72

The maximum product is 110

Calculating minimum product:

The largest positive and negative number is 11 and -9

11x-9=-99

The smallest positive and negative

-9x-8=-72

-99 is smaller than 72 so

maximum product =110, and minimum product=-99

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

Sol: Given key = 23 and

array = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91}

1. Initialize pointers

low = 0 and high = 9

Calculate $mid = \left[\frac{low + high}{2}\right] = \left[\frac{0+9}{2}\right] = 4$

compare arr[mid] with key:

arr[4] = 16

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

calculate $mid = \left[\frac{low + high}{2}\right] = \left[\frac{5+9}{2}\right] = 7$

Compute arr[mid] with key

arr[7] = 56

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

$mid = \left[\frac{5+6}{2}\right] = 5$

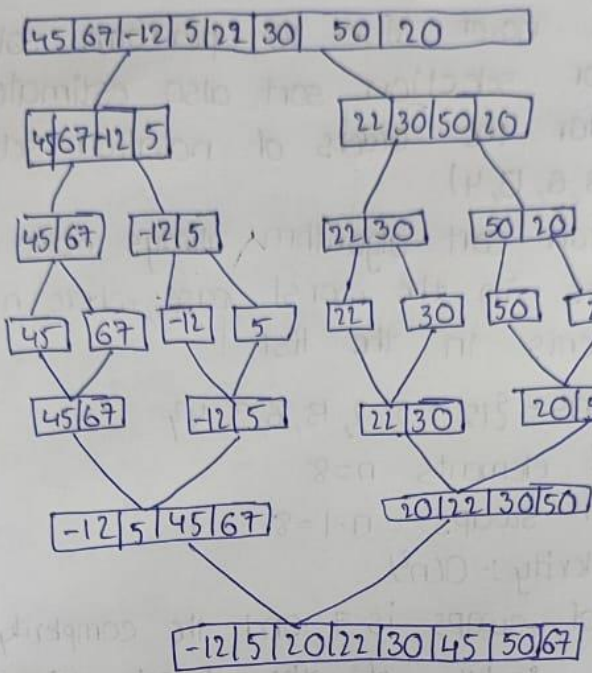
arr[mid] = arr[5] = 23

23 = 23 The key is found at index 5

∴ key = 23 is found at index 5.

3. Apply merge sort and other list of 8 elements Data d = [45, 67, -12, 5, 22, 30, 50, 20]. Set up recurrence relation for the numbers of key comparision made by merge sort.

Sol:



∴ Sorted list = {-12, 5, 20, 22, 30, 45, 50, 67}

4. Find the no.of times to perform supplying for selection sort estimate the time.

Recurrence relation for comparisons:

$$T(n) = 2T(n/2) + O(n)$$

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

→ At each level of recursion most n-1 comparisons to merge two six n so it becomes

$$T(n) = 2T(n/2) + (n-1)$$

Solving recurrence relation we get

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

$$\therefore T(n) = O(n \log n)$$

The recurrence relation is $T(n) = 2T(n/2) + O(n)$

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

ii Find the no.of times to perform solving suppling for selection sort also estimate time complexity for the orders of notation sets (12, 7, 5, -2, 18, 6, 13, 4)

The selection sort algorithm always makes exactly n-1 swaps in the worst case, where n is the no. of elements in the list

givens = {12, 7, 5, -2, 18, 6, 3, 4}

No.of elements $n=8$

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

Time complexity :- $O(n^2)$

The no.of swaps is 7, and the complexity is $O(n^2)$

5. Find the index of the target value 10 using binary search for following of value = 10

low = 0 and high = 9

$mid = \frac{low + high}{2} = \frac{0+9}{2} = 4$

list[4] = mid 10 Mid = value

Since 10 == 0 Mid = 10 . Mid = value

Sin 10 == 10 the target is found at index 4

∴ The target value = 10 is found at index 4