① Given an array of {4,-2,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9} integ
find the maximum and minimum product that can be obtained by multiplying two integers from the array

Sol: array is {4,-2,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,6,-8,11,-9}

We need to consider the largest and smallest products 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 product:

- The two largest positive numbers are 10 and 11

~~-9x-8=72~~ 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 on -9

11x-9=-99

The smallest ~~po~~ negative numbers are

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

compare 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

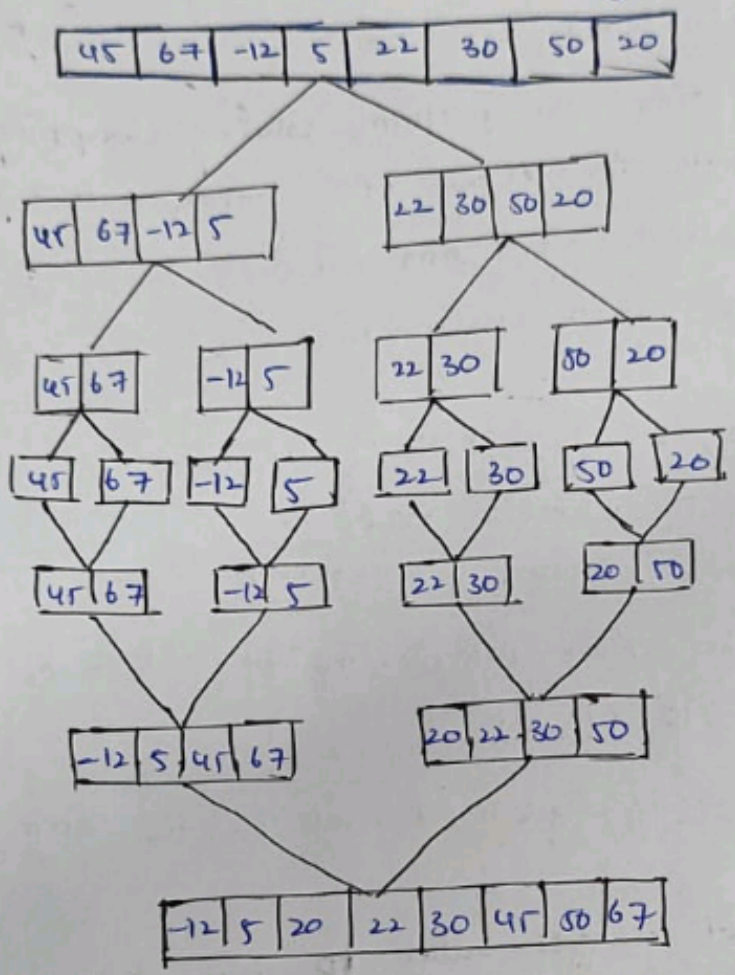
The key is found at index 5

∴ The key = 23 is found at index 5.

(3) Apply merge sort and otha list of 8 elements, data d = d = [45, 67, 12, 5, 22, 30, 50, 70). set up a recurrate relation for the number of key comparisons made by merge sort

sol: Merge sort.

given d = [45, 67, -12, 5, 22, 30, 50, 20]



∴ The sorted list = [-12, 5, 20, 22, 30, 45, 50, 67]

(4) Find the no. of times to perform swapping for selection sort also estimate the time

Recurrsive relation for comparision:

$$T(n) = 2 + (n/2) + O(n) \text{ if } n=1, T(1)=0$$

→ Each level of recursion we move at most n-1 comparison to merge two levels of size n so it becomes

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

solving recurrence 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

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

Find the no. of times to perform solving swapping for selection sort also estimate the time of notation set s (12, 7, 5, -2, 18, 6, 13, 14)

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

given $S = \{12, 7, 5, -2, 18, 6, 13, 14\}$

No. of elements $n = 8$

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

Time complexity: The time complexity of selection sort in Big-O notation is $O(n^2)$

So, the number of swaps is 7 and and the time complexity is $O(n^2)$

(5) Find the index of target value to using binary search from following list of elements [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Sol: given list = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20] and value = 10

low = 0 and high = 9

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

mid = 10 mid = value

since 10 = 10 the target is found at index 4

∴ The target value = 10 is found at index 4