1) Given an array of [4,-2,5,3,10,-5,2,8,-3,6,7,-4,1
9,-1,10,-6,8,11,9] integers and the maximum and minimum product that can be obtained by multiply two integers from the array.

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

we need to consider the largest and smallest product that can be formed by selecting two consider 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 = 110
* the two smallest negative numbers are -9 and -8 = 72

The maximum product is 0

Calculating minimum product:

The largest positive and negative number 11 and -9

11 x -9 = -99

The smallest positive and negative numbers are

-9 x 8 = -72

-99 is smaller than 72 : 0

maximum product = 110, and minimum product = -99

2) Demonstrate the priority search method to search for the key = 23 from the array = {2, 5, 8, 12, 16, 25, 38, 56, 72, 91}

Sol)

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

1. intalize pointers

low = 0 and high = 9

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

Compare arr[mid] with key:

arr[4] = 16

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

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

Compare arr(mid) with key:

arr[7] = 56

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

mid = $\left(\frac{5+6}{2}\right) = 5$

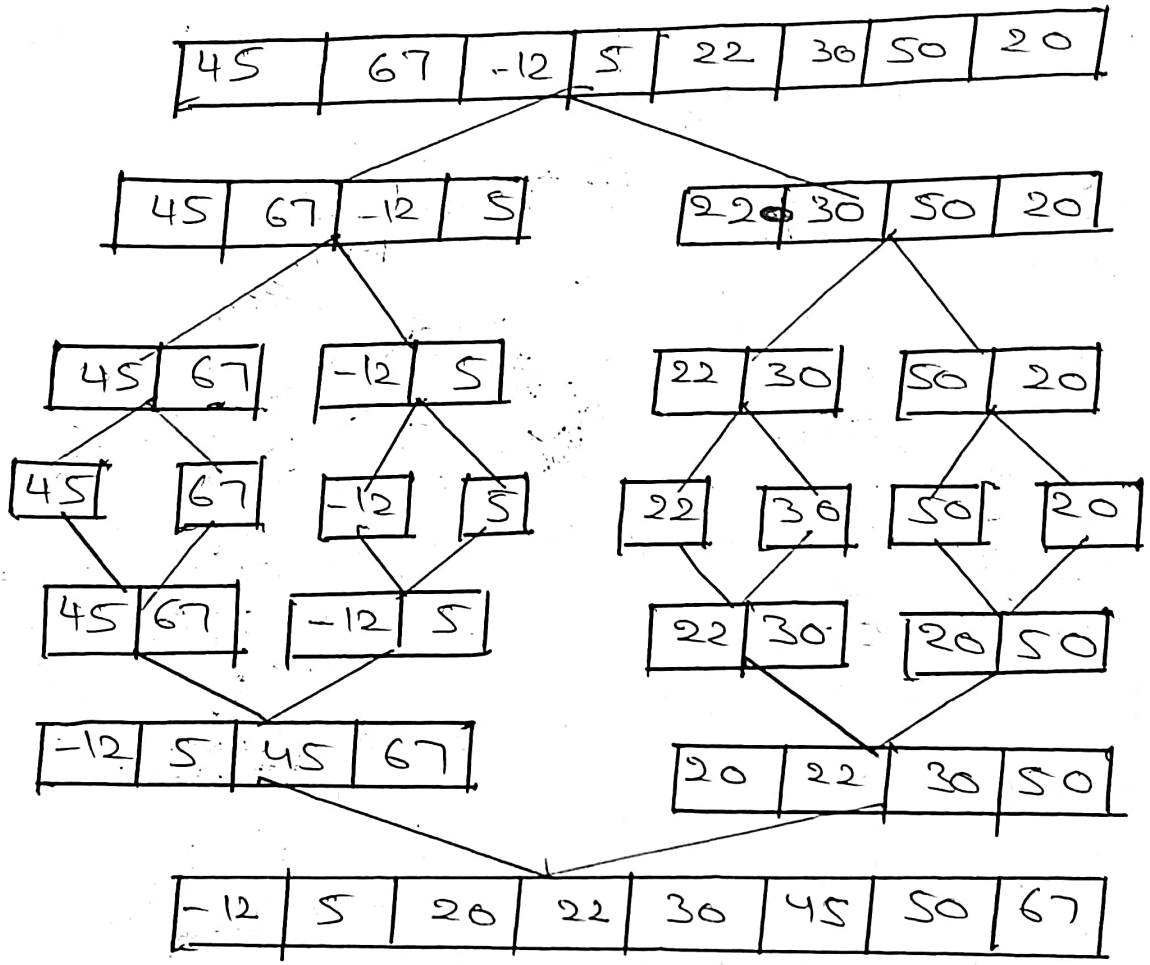
arr[mid] = arr[5] = 23

23 = = 23 The is found at index 5

∴ The 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 a recursive relation for the number of key comparisions made by merge sort

3) Merge sort



∴ The sorted list = (-12, 5, 20, 22, 30, 45, 50, 67)

4) Find the times of Recurrence Relation for Comparision:

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

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

Find the no. of times to perform solving swapping for selection sort also estimate the time complexity for the order of notation set S(12, 7, 5, -2, 18, 6, 13, 4)

Sol) The selection sort algorithm always makes exactly n-1 swaps in the worst case (where n is the no. of element in the list

given S = {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 notation is $O(n^2)$

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

(5) Find the index of the target value 10 using binary search from the 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

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

Ex: List[4] · mid = 10. mid = = value

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

∴ The Target value = 10 is found at index 4.