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

Given 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 the numbers from the array

1. Sort the array.

   sorted array is [-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 integers are 10 and 11

$10 \times 11 = 110.$

→ The two smallest negative integers are -9 and -8

$-9 \times -8 = 72,$

→ The maximum product is 110.

Calculating minimum products

→ largest positive and negative number is 11 and -9

$11 \times -9 = 99$

→ The smaller negative numbers are

$-9 \times -8 = 72.$

-99 is smaller than 72 so.

maximum product = 110.

and minimum product = -99.

(12) Demonstrate the binary search method to search for the =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 $mid = \left\lfloor \frac{low + high}{2} \right\rfloor = \left\lfloor \frac{0+9}{2} \right\rfloor = 4$

Compare arr(mid) with key:

$arr(4) = 16$

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

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

Compare arr(mid) with key:

$arr(7) = 56$

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

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

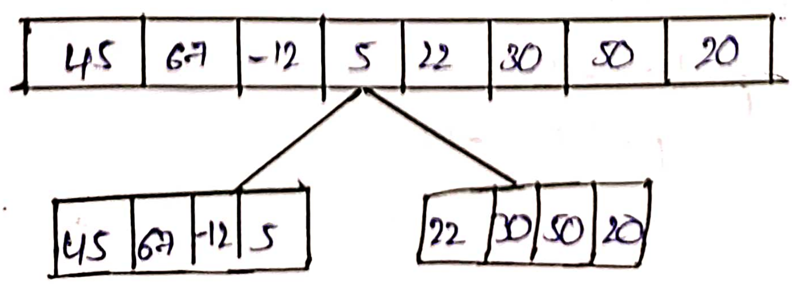
$arr(mid) = arr(5) = 23.$

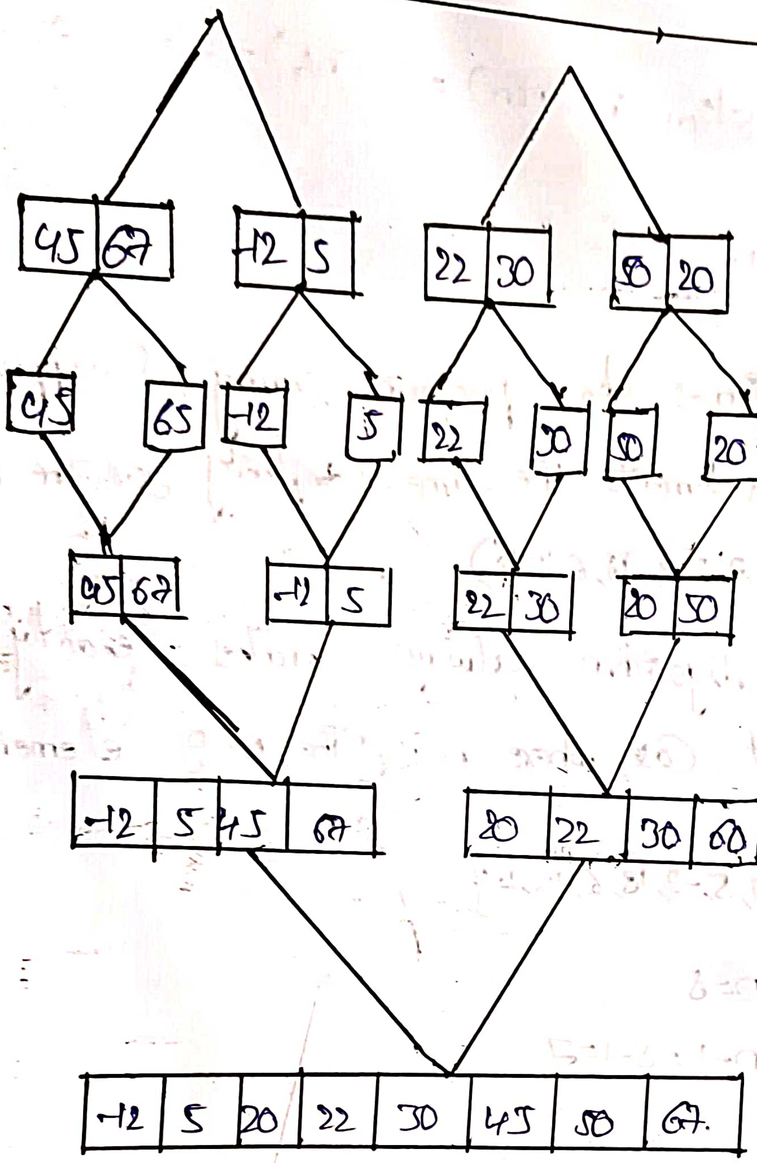
$23 == 23$ the key is found.

∴ The key = 23 is found at index 5.

(13) 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 number of key comparisions made by merge.

Sol:





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

Recurrence Relation for comparisons:

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

If $n=1$, $T(1)=0$ Base case.

→ At each level of Recursion we make at most $n-1$ comparision to merge two halves of size $n/2$ so it becomes.

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

Solving Recurrence Relation we get

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

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

$\therefore$ The Recurrence Relation is $T(n) = 2T(n/2) + O(n)$ or more precisely.

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

(4) find the no. of times to perform solving swapping for selection sort also asymptotes the time Complexity for the orders of rotation sets (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 elements 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 number of swaps is 7, and the time Complexity is $O(n^2)$.

(15) 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].

A: Given list = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20} and value = 10.

Low = 0 and high

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

List(4) = mid = 10 mid == value.

Since 10 == 10 The target is found at index.

∴ The target value = 10 is found at index.