11. Given an array of {4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, -9} integer. Find the max and min product that can be obtained by multiplying two integers from the array.

sol:- 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 Product :-

* The two largest positive numbers are 10 and 11 $\Rightarrow$ $10 \times 11 = 110$
* The two smallest negative numbers are -9 and -8 $\Rightarrow$ $-9 \times -8 = 72$

The maximum product is 110

Calculating minimum product :

The largest positive and negative number is 11 on -9 $\Rightarrow$ $11 \times -9 = -99$

The smallest negative number is $-9 \times -8 = 72$

-99 is smaller than 72 So,

Maximum product = 110 and minimum product is -99

---

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

sol:- ① Initialize Pointers.

low = 0 and high = 9 i.e.

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 $high = mid - 1 = 6$

$mid = \left\lfloor \frac{5+6}{2} \right\rfloor = 5$

$arr[mid] = arr(5) = 23$

$23 == 23$

The key is found at index 5

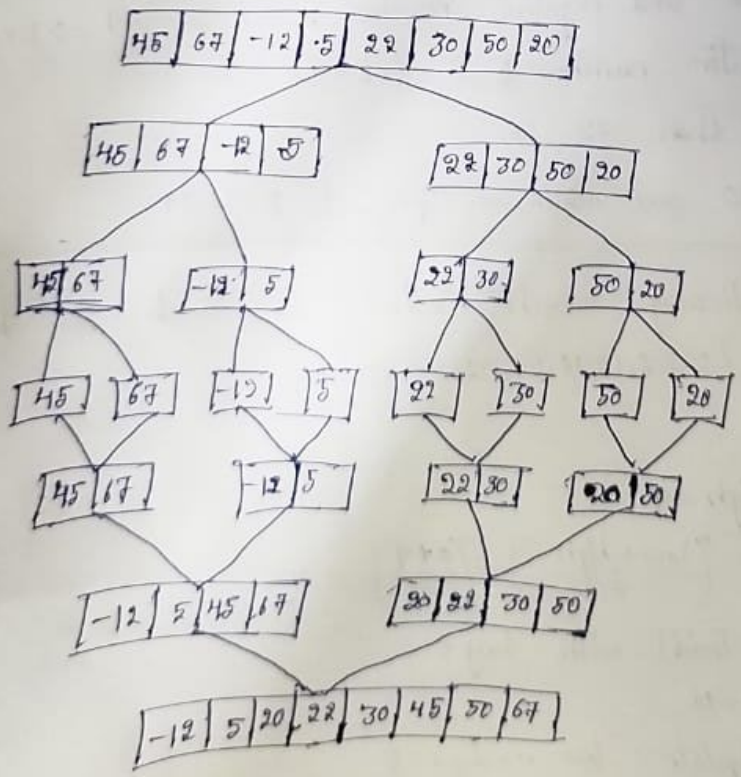
∴ The key = 23 is found at index 5

---

12. Apply merge sort and other list of 8 elements data $d = [45, 67, 12, 5, 22, 30, 50, 20]$. Set up a recurrate relation for the no. of key compare 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]

14. Find the no.of times to perform swapping for selection sort also estimate the time recursive relation for comparison

Sol:- $T(n) = 2 + (n/2) + O(n)$. if $n=1$; $T(1) = 0$

All each level of recursion we move at most $n-1$ comparison to merge two level of size n so it because

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

solving recurrence relation we get

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

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

The recurrence relation is $T(n) = 2T(n/2) + C(n)$. or more precisely.

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

15. Find the no.of times to perform solving for selection sort also estimate the time of notation sets $(12, 7, 5, 2, 18, 6, 13, 4)$. Find the index of target value to using binary search from following list of elements $[2, 4, 6, 8, 10, 2, 14, 16, 18, 20]$.

sol:- Given list = $[2, 4, 6, 8, 10, 2, 14, 16, 18, 20]$ and value = 10

low = 0 and high = 9

$$mid = \left\lceil \frac{low + high}{2} \right\rceil = \left\lceil \frac{0+9}{2} \right\rceil = 4$$

mid = 10 mid = value

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

∴ The target value = 10 is found at index 4

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 noof swaps is 7 and the time Complexity is $O(n^2)$.