1. Given an array of [4,-2,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11] integers 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]

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 = 10x11=110
* The two smallest negative numbers are -9 and -8 = 72

The maximum product is 110

calculating minimum products

The largest positive and negative numbers is 11,-9 = -99

The smallest positive and negative numbers -9x-8 = 72

2. Demonstrate the primary 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 & array = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91}

* Initialise pointers

low = 0 and high = 9

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

compare arr(mid) with key:

arr(4) = 16

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

calculate mid = $\frac{low + high}{2} = \frac{5+9}{2} = 7$

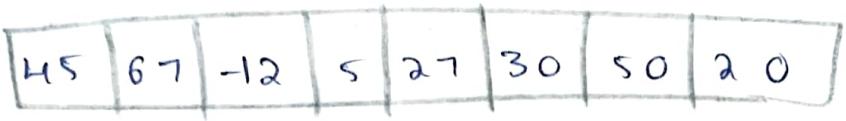
arr[7] = 56 56 > 23 update high = mid - 1 = 6

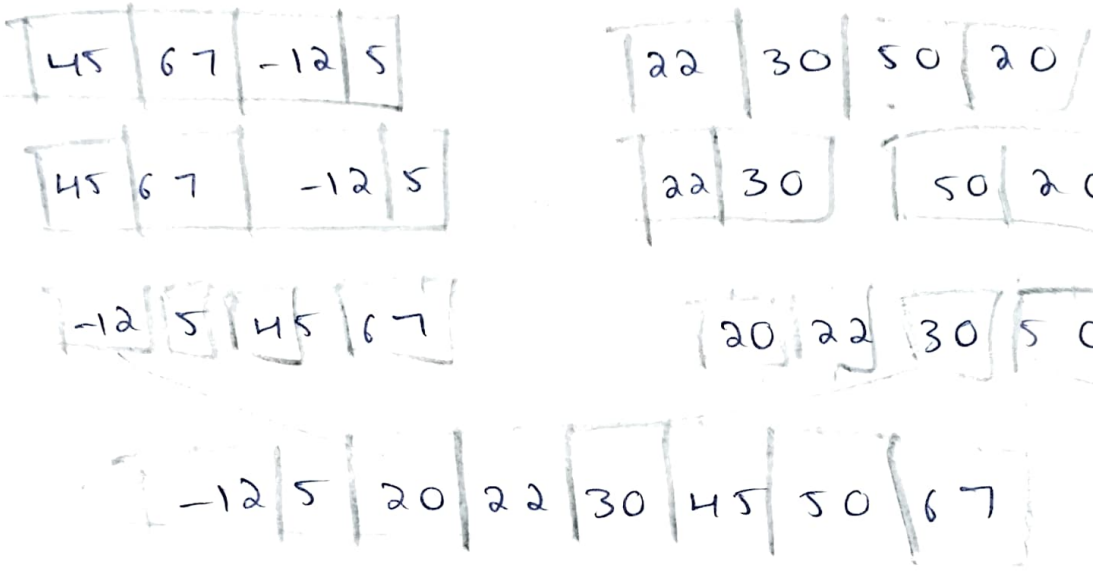
mid = $\left(\frac{5+6}{2}\right) = 5$ => arr [mid] = arr[5] = 23; 23 = 23

The key is found at index 5 The key = 23 is found at index 5

3. Apply merge sort and other list of 8 elements, Data d = (45, 67, -12, 5, 22, 30, 50, 20). Set up a recurrence relation for the no. of key comparisons made by merge sort.

Sol) Merge sort:-





∴ The sorted list = (-12, 5, 20, 23, 30, 45, 50, 67)

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

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

→ At each level of recursion we make at most in 1 comparison merge two halves of six n/2 So it becomes $T(n) = 2T(n/2) + (n-1)$

solving recurrence relation we get

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

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

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

4. Find the no.of times to perform solving swapping for sel. sort also estimate the time complexity for others of notations sets (12,7,5,-2,18,6,13,-7)

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.

S: {12,7,5,-2,18,6,13,44}

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 no.of swap and time complexity is $O(n^2)$.

5. Find the index of the target value to visting binary search team the following list of elements {2,4,6,8,10,12,14, 10,20}

Sol. Given, List {2,4,1,8,10,2,14,16,18,20} and route=10

low=0 and high=9

$$mid = \frac{l+h}{2} = \frac{0+9}{2} = 4$$

mid=10, mid == value

since==10 the target is found at index 4

∴ The target value=10 is formed at index 4