M Vishnu Vardhan Reddy
192325061

8) 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.

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

The maximum Product is 110.

calculating minimum Product's
The largest positive and negative number is 11 and -9.
$11 \times -9 = 99$
The smallest negative numbers are -9 and -8
$-9 \times -8 = 72$
-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 = $\left[\frac{Low+high}{2}\right] = \left[\frac{5+9}{2}\right] = 7$

compute arr[mid] with key

arr[7]=56

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

mid = $\left[\frac{5+6}{2}\right] = 5$

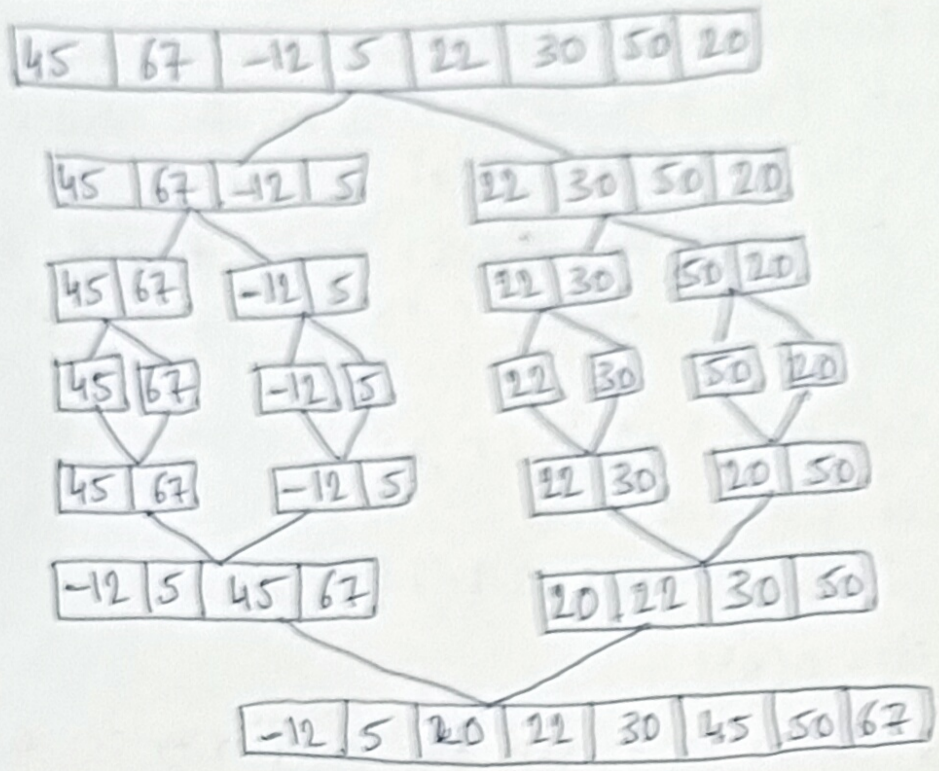
arr[mid] = arr[5] = 23

23==23 the key is found at index 5.

∴ The key=23 is found at index 5.

3) Apply merge sort and otherlist of 8 elements, Data d=(45, 67, -12, 5, 22, 30, 50, 20). set up recurrece relation for the numbers of key comparision made by merge sort

Sol.



the sorted list = {-12, 5, 20, 22, 30, 45, 50, 67}

4) find the no of times to perform supplying for selection sort estimate the time.

Recurrence relation for comparsions:

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

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

→ At each level of Recursion most $n-1$ comparsions to merge two of size n so it becomes

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

Solving recurrence relation we get

$$T(n) = n \log 2(n) + n + 1$$

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

the recurrence relation is $T(n) = 2T(n/2) + O(n)$

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

4) find the no of times to perform solving swapping for selection sort also estimate the time complexity for the orders of notation sets (12,7,5,-2,18,6,13,4)

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 = $O(n^2)$

the no of swaps is 7 and the complexity is $O(n^2)$

5) find the index of the target values 10 using binary solution search for the following list of elements [2,4,6,8,10,2,14,16,18,20] given list[2,4,6,8,10,2,14,16,18,20] and value = 10.

Low = 0, and high = 9

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

List[4] = mid = 10 mid = Value.

since to = = 10 the target is found at index 4.

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