ASSIGNMENT → 4

Name of the Student : A. Manoj Reddy

Name of the course : Design for analysis of algorithm

Register NO : 192311171

Course Code : CSA0670

Date of Submission :

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} inter. find the max and min product that can be obtained by multing two integer from the array.

Sol:- We need to consider the loggest and smallest, Product that can be formed by selecting two number from the array.

① Sort the array

{-9, -8, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

② identify Possible consider for maximum product.
③ identify Possible conditetes for minimum product.

Calculating maximum product:-

* The two largest positive numbers are 10 and 11 =>
  10 × 11 = 110
* The two smallest negative numbers are -9 and -8 =>
  -9 × -8 = 72

The maximum product is 110

Calculating minimum Product:

The larget positive and negative number is 11 on -9 =>
11 × -9 = -99

The smallest negative number is -9 × -8 = 72

-99 is smaller then 72 so,

maximum product = 110. and minimum product is -99.

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

① Initialize pointers
low = 0 and high = 9

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

Compare arr [mid] with key:

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

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

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

The key is formard at index 5

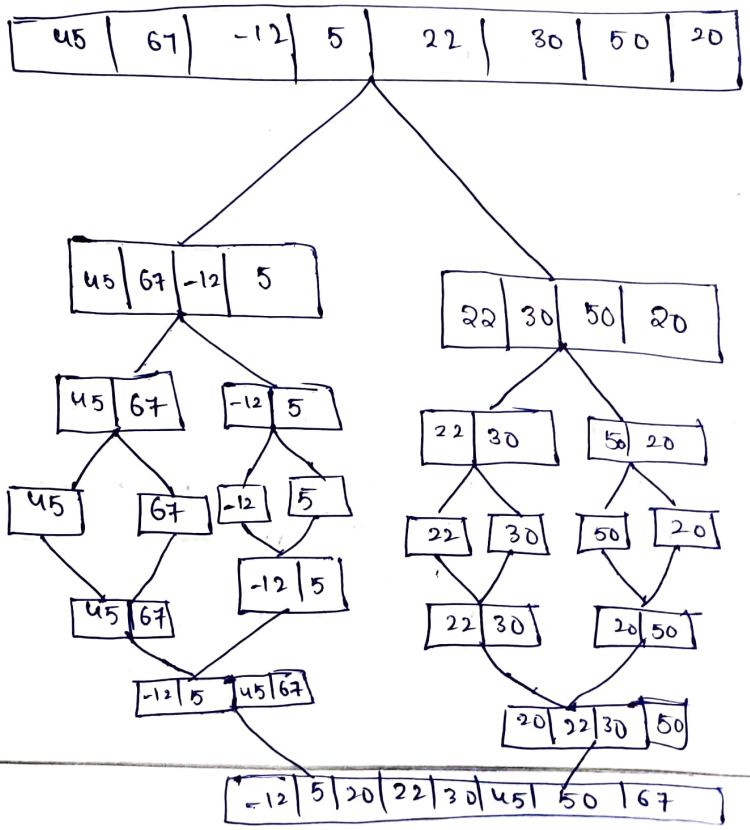
∴ The key = 23 is found at index 5

13. Apply merge sort and other list of 8 elements data d = [45, 67, 12, 5, 22, 30, 50, 70]. Setup a reccurate relation for the no. of key compare made by merge sort.

Sol:

merge sort:

Given d = [45, 67, -12, 5, 22, 30, 50, 20]



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

$T(n) = 2 + (n/2) + O(n)$ if $n = 1$; $T(1) = 0$

b) Find the no. of times to perform solving for selection sort also estimate the time of notation sets [12,7,5,-2,18,6,3,4]

- find

Sol) a) Given, At each level of recursion we more at most $n-1$ comparision to merge two level of size n) so it because $T(n) = 2 + (n/2) + (n-1)$

Solving recurrence relation we get

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

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

The recurrance relation is $T(n) = 2T(n/2) + O(n)$ or more precisely $T(n) = n \log_2(n) - n + 1$

b) Given list = [2, 4, 6, 8, 1]

Given S = [12, 7, 5, -2, 18, 6, 3, 4]

No. of elements $n = 8$

No of swaps $= n-1 = 8-1 = 7$

No. of swaps $= n-1 = 8-1 = 7$

Time Complexity:

The time complexity of selection sort in dig O-notation is $O(n^2)$ so, the no. of swaps is 7 and the time complexity is $O(n^2)$.