Assignment - 05

Name : M. Hiteshwar Reddy

Reg no : 19232507 4

Dept : AIML

Course Code : CSA0389

Course Name : Data structure

Faculty Name : Dr. Ashok kumar.

Date of submission : 21-08-24

No. of pages : 06

1) i) Write the algorithm for insertion sort the following Sequences 3, 1, 4, 1, 5, 9, 2, 6, 5

ii) Explain the procedure for manage sort & perform merge sort for following inputs. Also show the result for each step of iteration 64, 8, 216, 512, 27, 729, 0, 1, 343, 125

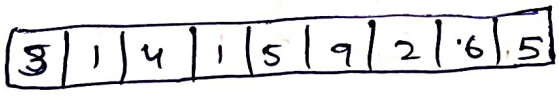
i) Algorithm

```
insertion sort (a,n);
  for (i=1, i<n-1; i++)
     key = a[i]
     j = i-1
     while j>0 & a[i]> key
       a[j+i] = a[j]
       j = j-1
       a[j+1] = key
```

Sorting :-

Initialize temp variable

temp

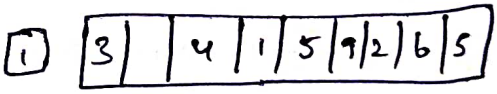


Step 1 :-

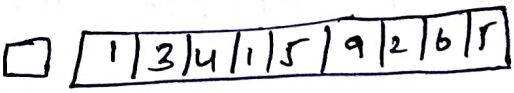
a[0] = 3, a[1] = 1

a[0] > a[1]

1 goes to temp

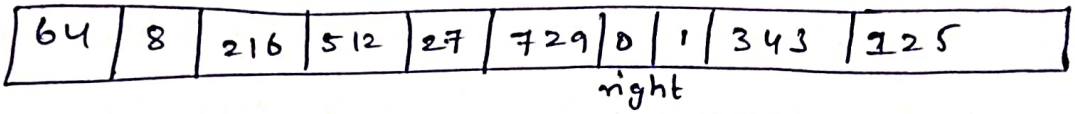


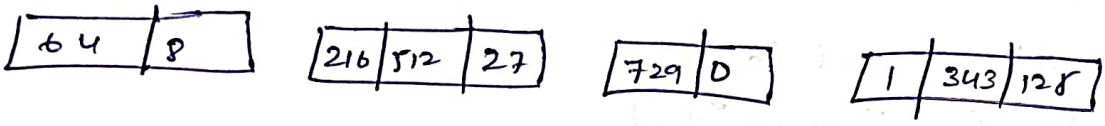
1 < 3, a[0] = 1

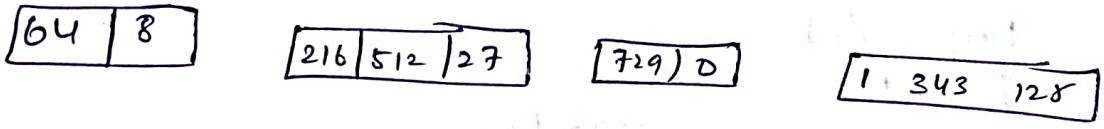


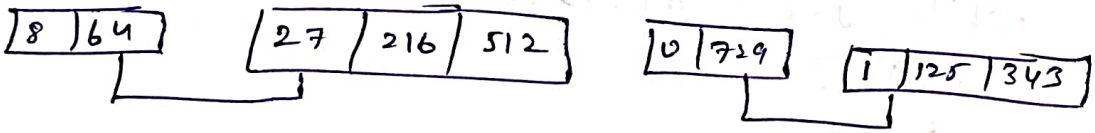
ii) Merge sort

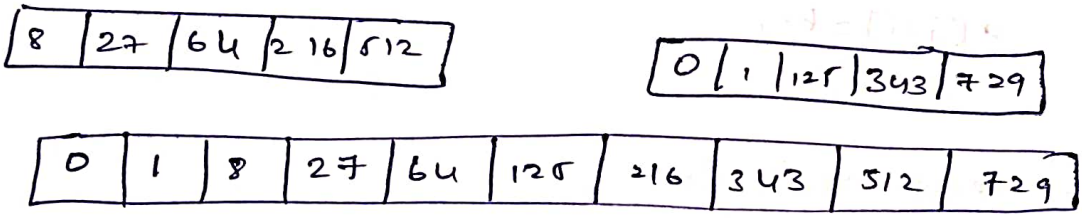
Initial array: [64, 8, 216, 512, 27, 729, 0, 1, 343, 125]











2) Draw concept map of partioning in quick sort, try to write an algorithm for it, which is follow, Develop a program considering these steps.

Step1:- choose the highest index value has pivot

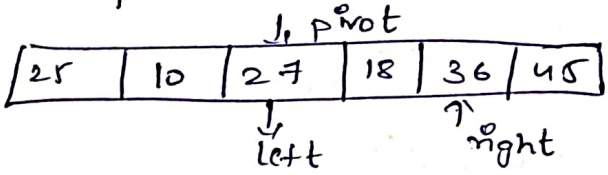
Step-2:- Take two variables to point left & right of the list excluding pivot.

Step-3:- Left points to the low index.

Using elements your own

a[left]=36, a[pivot]=0 [right]=27, a[pivot] = 0[left]

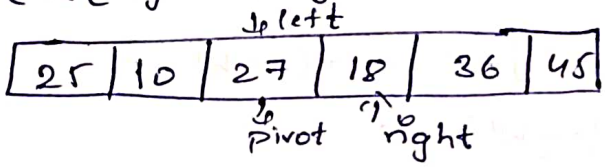
so swap



a [left] = a [pivot] = 27, a[right] = 36.

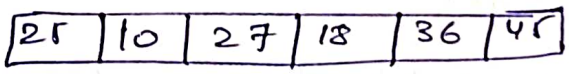
→ Since pivot is at left, so algorithm starts from right & move to left.

a[pivot] < a [right], right moves one position forward



a [left] = a [pivot] = 27 , a[right] = 18

a [pivot] > a[right] so swap



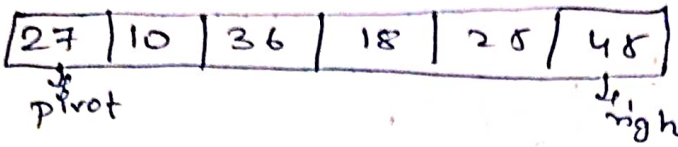
→ Since pivot is at right so algorithm starts from left & moves to right.

a [left] = 18 a [pivot] = a[right] = 27

a[pivot] > o[left] . so left moves one posn forward

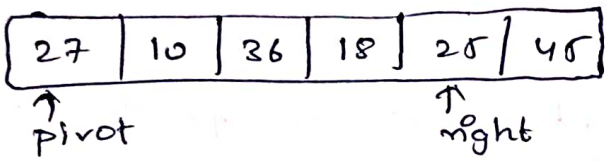
→ Now, o [pivot], a[left] & a[right] are same, so there are pointing the same element, it represents the termination of procedure.

a[] = {27, 10, 36, 18, 25, 43}



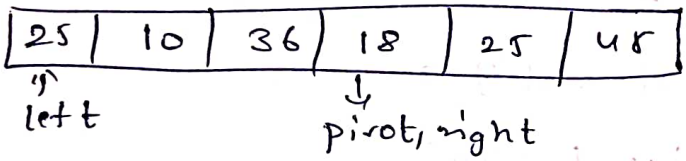
Compare a[pivot] & a[right]

a[pivot] < a[right] . so right moves forward one position



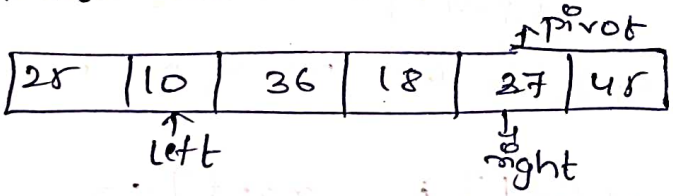
a[left] = a[pivot] = 27 , a[right] = 25.

a[pivot] > a[right] , so swap



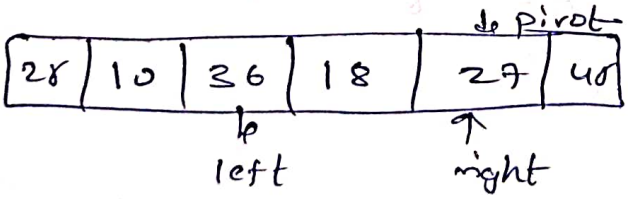
→ since , pivot is at right , so algorithm starts from left & moves to right.

a[pivot] > a[left] . so algo moves one position to right



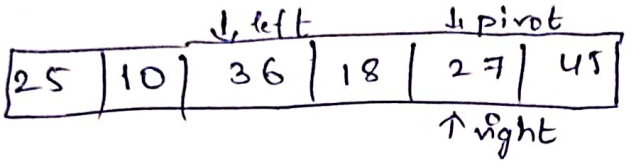
a[left] = 10, a[pivot] = a[right] = 27

a[left] < a[pivot], so left moves forward



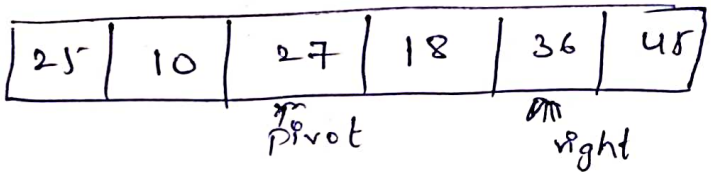
a[left] = 10, a[pivot] = a[right] = 27

a[left] < a[pivot], so left moves forward



a[left] = 36, a[pivot] = a[right] = 27, a[pivot] < a[left]

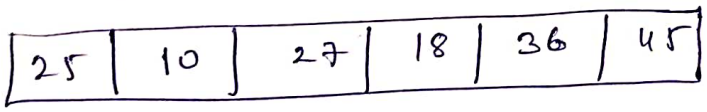
So Swap



a[left] = a[pivot] = 27, a[right] = 36

* Since pivot in at left, so algorithm starts from right, more to left.

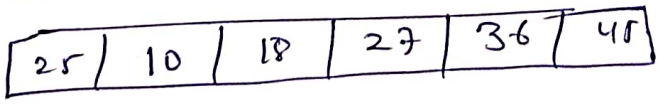
a[pivot] < a[right] right moves one posn forward



a[left] = 18 a[pivot] = a[right] = 27

a[pivot] > a[left], so left moves one position forward.

* Elements that are right side of element 27 greater than it & elements that are left side of 24 are smaller



* Now in a similar manner, quick sort algorithm is separately applied to left & right subarrays.