11-07-2024

INSERTION SORT:-

```
-> main()
{
    int a[20], i, n;
    printf("Enter n : ");
    scanf("%d", &n);
    printf("Enter elements :");
    for(i=0; i<n; i++)
    scanf("%d", &a[i]);
    insertion_sort(a, n);
    for(i=0; i<n; i++)
    printf("%d", a[i]);
}
void insertion_sort(int a[], int n)
{
```

```
int b[20], i, j;
b[0] = a[0];
for (i = 1; i < n; i++)
{
  for (j = i-1; a[i] < b[j] && j >= 0; j--)
  {
    b[j+1] = b[j];
  }
  b[j+1] = a[i];
}
for (i = 0; i < n; i++)
  a[i] = b[i];
}
```

~~Nov~~ 2024

QUICK SORT :-

PROCEDURE =>

→ select 1st element as i

→ select 1st element as Pivot

→ select last element as j

→ Increase i until a[i] <= a[P], i++

→ Decrease j until a[j] > a[P], j--

→ If i, j donot crosses eachother swap a[i] & a[j]

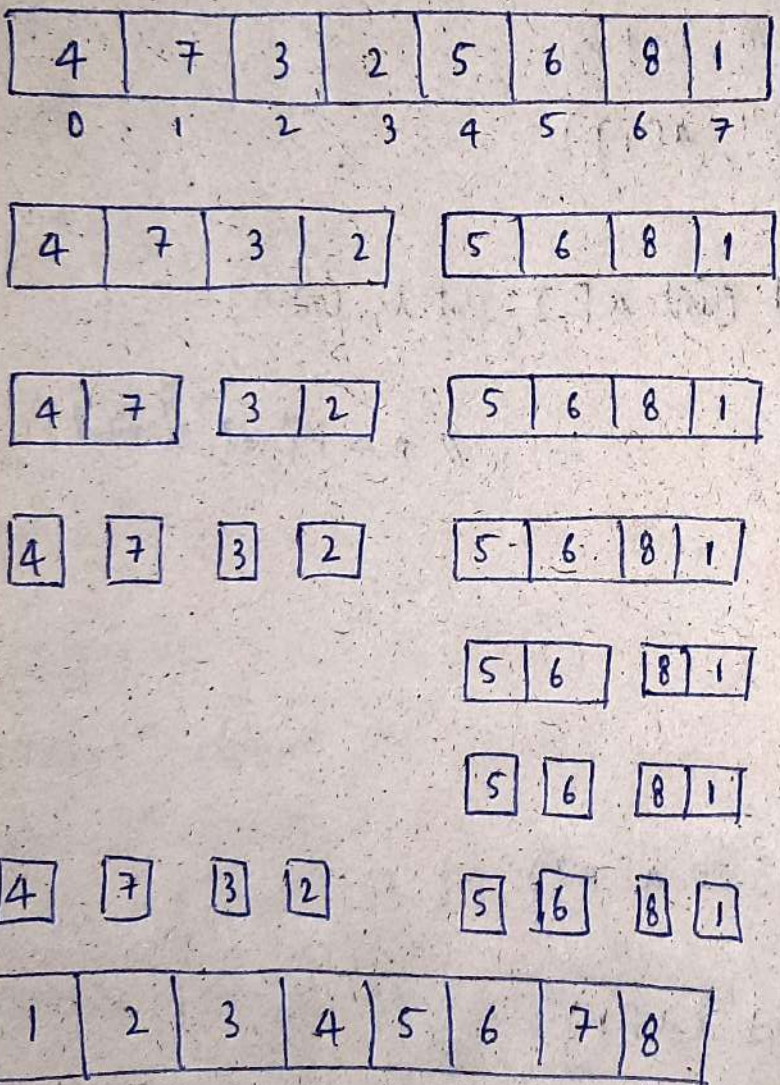
→ Repeat 4 & 5 steps until i crosses j

→ If i crosses j swap a[j] & a[P].

```
-> main ()
{
  int a[20], n, i;
  printf ("Enter n: ");
  scanf ("%d", &n);
  printf ("Enter elements: ");
  for (i=0; i<n; i++)
  scanf ("%d", &a[i]);
  quick_sort (a, 0, n-1)
  for (i=0; i<n; i++)
  printf ("%d", a[i]);
}
void quick_sort (int a[], int l, int h)
{
  int i, j, P, t;          // P -> Pivot element
  if (l<h)
  {
    i = P = l;
    j = h;
    while (a[i] <= a[P])
    i++;
    while (a[j] > a[P])
    j--;
    if (i<j)
    {
      t = a[i];
      a[i] = a[j];
      a[j] = t;
    }
}
```

```
    t = a[j];
    a[j] = a[p];
    a[p] = t;
    quick_sort(a, l, j-1);
    quick_sort(a, j+1, h);
    }
}
```

MERGE SORT:-



```
-> main ()
{ int a[20], i, n;
  printf ("Enter n: ");
  scanf ("%d", &n);
  printf ("Enter elements: ");
  for (i=0; i<n; i++)
  scanf ("%d", &a[i]);
  merge-sort (a, 0, n-1);
  for (i=0; i<n; i++)
  printf ("%d", a[i]);
}
void merge_sort (int a[], int l, int m, int n)
{
  int b[20];
  int i=l, j=m+1, k=l;
  while (i<=m && j<=h)
  {
    if (a[i] <= a[j])
    {
      b[k] = a[i];
      i++;
    }
    else
    {
      b[k] = a[j];
      j++;
    }
  k++;
}
```

```
while (i<=m)
{
    b[k] = a[i];
    i++;
    k++;
}
while (j<=h)
{
    b[k] = a[j];
    j++;
    k++;
}
for(i=l; i<=h; i++)
{
    a[i] = b[i];
}
}
```