ASSIGNMENT – 10

**Session 1**

1. write a Algorithm of bubble sort.

**package** my.sorting;

**import** java.util.Arrays;

**public** **class** bubble {

**public** **static** **void** main(String[] args) {

**int**[] a= {4,1,5,2,3};

System.***out***.println("Array before sort-- "+Arrays.*toString*(a));

**int** n=a.length;

**for**(**int** i=0;i<n-1;i++) {

**for**(**int** j =0;j<n-1;j++) {

**if**(a[j]>a[j+1]) {

**int** tmp=a[j];

a[j]=a[j+1];

a[j+1]=tmp;

}

}

}

System.***out***.println("Array after sorting-- "+Arrays.*toString*(a));

}

}

OUTPUT:

Array before sort-- [4, 1, 5, 2, 3]

Array after sorting-- [1, 2, 3, 4, 5]

1. Write a Algorithm of merge sort .

**package** my.sorting;

**public** **class** merge {

**public** **static** **void** merge(**int** a[],**int** l,**int** m,**int** r) {

**int** n1=m-l+1;

**int** n2=r-m;

**int** L[]=**new** **int**[n1];

**int** R[]=**new** **int**[n2];

**for**(**int** i=0;i<n1;i++) {

L[i]=a[l+i];

}

**for**(**int** j=0;j<n2;j++) {

R[j]=a[m+1+j];

}

**int** i=0,j=0,k=l;

**while**(i<n1 && j<n2) {

**if**(L[i]<=R[j]) {

a[k]=L[i];

i++;

}

**else** {

a[k]=R[j];

j++;

}

k++;

}

**while**(i<n1) {

a[k]=L[i];

i++;

k++;

}

**while**(j<n2) {

a[k]=R[j];

j++;

k++;

}

}

**public** **static** **void** sort(**int** a[] , **int** l,**int** r) {

**if**(l<r) {

**int** m=(l+(r-1))/2;

*sort*(a,l,m);

*sort*(a,m+1,r);

*merge*(a,l,m,r);

}

}

**public** **static** **void** display(**int** a[]) {

**int** n=a.length;

**for** (**int** i=0;i<n;++i) {

System.***out***.print(a[i]+" ");

}

System.***out***.println();

}

**public** **static** **void** main(String[] args) {

**int** a[]= {3,60,43,27,6,19,32,50,4,90};

System.***out***.println("Array Before using merge Sort:-");

*display*(a);

*sort*(a,0,a.length-1);

System.***out***.println("Array After using merge Sort:-");

*display*(a);

}

}

OUTPUT:

Array Before using merge Sort:- 3 60 43 27 6 19 32 50 4 90

Array After using merge Sort:- 3 4 6 19 27 32 43 50 60 90

1. Write a Algorithm to  quicksort .

**package** my.sorting;

**public** **class** quick {

**public** **static** **void** swap(**int** a[] , **int** t1 , **int** t2 ) {

**int** temp=a[t1];

a[t1]=a[t2];

a[t2]=temp;

}

**public** **static** **int** part(**int** a[], **int** l,**int** h) {

**int** p=a[h];

**int** s=l-1;

**for** (**int** j=l;j<h;j++) {

**if**(a[j]<p) {

s++;

*swap*(a,s,j);

}

}

*swap*(a,s+1,h);

**return** (s+1);

}

**public** **static** **void** sort(**int** a[] , **int** l,**int** h) {

**if**(l<h) {

**int** pi=*part*(a,l,h);

*sort*(a,l,pi-1);

*sort*(a,pi+1,h);

}

}

**public** **static** **void** display(**int** a[]) {

**int** n=a.length;

**for**(**int** i=0;i<n;i++) {

System.***out***.print(a[i]+" ");

}

System.***out***.println();

}

**public** **static** **void** main(String[] args) {

**int**[] a= {3,48,67,23,45,98,65,27,5,70};

System.***out***.println("Array before Quick Sort-->");

**int** n=a.length;

*display*(a);

*sort*(a,0,n-1);

System.***out***.println("Array After Quick Sort-->");

*display*(a);

}

}

OUTPUT:

Array before Quick Sort--> 3 48 67 23 45 98 65 27 5 70

Array After Quick Sort--> 3 5 23 27 45 48 65 67 70 98