1. Linear Search
2. **package** DSA;
3. **public** **class** LinearSearch {
4. **public** **static** **void** main(String[] args) {
5. **int** a[] = {10, 20, 35, 53, 70, 42};
6. **int** size = a.length;
7. **int** v = 53;
8. **for**(**int** i=0;i<size-1;i++) {
9. **if**(a[i]==v) {
10. System.***out***.println("element found index is:"+i);
11. }
13. }
14. }

17.}

2.Binary search

**package** DSA;

**public** **class** binarysearch {

**public** **static** **void** binary(**int** a[], **int** f, **int** l, **int** k) {

**int** mid =(f+l)/2;

**while**(f<=l) {

**if**(a[mid]<k) {

f=mid+1;

}**else** **if**(a[mid]==k){

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

**break**;

}

**else** {

l = mid-1;

}

mid=(f+l)/2;

}

**if**(f>l) {

System.***out***.println("Element not found");

}

}

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

**int** a[] = {15,20,25,30,50};

**int** k = 25;

**int** l = a.length-1;

*binary*(a,0,l,k);

}

}

3. Insertion sort

**package** DSA;

**public** **class** Insertionsort {

**public** **static** **void** insertionsort(**int** arr[]) {

**int** N = arr.length;

**for**(**int** j=1;j<N;j++) {

**int** k = arr[j];

**int** i=j-1;

**while**((i>-1)&&(arr[i]>k)) {

arr[i+1]=arr[i];

i--;

}

arr[i+1] = k;

}

}

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

**int**[] a = {8, 12,4,6,28,76,33};

**for**(**int** i:a) {

System.***out***.print(i+" ");

}

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

*insertionsort*(a);

**for**(**int** i:a) {

System.***out***.print(i+" ");

}

}

}

4.Selection sort

**package** DSA;

**public** **class** selectionsort {

**void** sort(**int** a[]) {

**int** n = a.length;

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

**int** min\_element = i;

**for**(**int** j=i+1;j<n;j++)

**if**(a[j]<a[min\_element])

min\_element=j;

**int** temp = a[min\_element];

a[min\_element]=a[i];

a[i] =temp;

}

}

**void** printarray(**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) {

selectionsort s = **new** selectionsort();

**int** a[] = {10, 16, 5,55,37};

s.sort(a);

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

}

}

5.Bubble sort

**package** DSA;

**import** java.util.Scanner;

**public** **class** Bubblesort {

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

Scanner sc = **new** Scanner(System.***in***);

**int** n = 6;

**int**[] a = **new** **int**[n];

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

a[i] = sc.nextInt();

}

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

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

{

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

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

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

a[j+1]=x;

}

}

}

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

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

}

}

6. Merge sort

**package** DSA;

**import** java.util.Scanner;

**public** **class** Mergesort {

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

**int** i,j,c=1;

**int** b[]=**new** **int** [m+1];

**for**(i=1, j=n+1;i<=n &&j<=m; c++) {

**if**(a[i]<=a[j])

b[c]=a[i++];

**else**

b[c]=a[j++];

}

**while**(i<=n) {

b[c++]=a[i++];

**while**(j<=m)

b[c++]=a[j++];

}

**for**( i=1;i<=m;i++)

a[i] = b[i];

}

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

**if**(l<h) {

**int** k = (l+h)/2;

*sort*(a,l,k);

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

*merge*(a,l,k);

}

}

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

// **TODO** Auto-generated method stub

**int** n, r,i;

Scanner s = **new** Scanner(System.***in***);

n=s.nextInt();

**int** a[] = **new** **int**[n];

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

a[i] = s.nextInt();

}

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

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

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

}

}

7. Quick sort

**package** DSA;

**public** **class** Quicksort {

**int** partition(**int** arr[], **int** start,**int** end) {

**int** pivot = arr[end];

**int** i=(start-1);

**for**(**int** j=start;j<=end-1;j++) {

**if**(arr[j]<pivot) {

i++;

**int** t =arr[i];

arr[i]=arr[j];

arr[j]=t;

}

}

**int** t = arr[i+1];

arr[i+1] = arr[end];

arr[end] = t;

**return**(i+1);

}

**void** quick(**int** a[], **int** start, **int** end) {

**if**(start<end) {

**int** p = partition(a,start,end);

quick(a,p+1,end);

}

}

**void** print(**int** a[], **int** n) {

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

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

}

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

// **TODO** Auto-generated method stub

**int** a[] = {13, 2, 17, 25, 12, 27};

**int** n=a.length;

Quicksort q1 = **new** Quicksort();

q1.print(a,n);

q1.quick(a, 0, n-1);

q1.print(a,n);

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

}

}