Linear search:

Algorithm:

Step-1: Set the key, starting index (low), ending index (end)

Step-2: for low to end

1. If key == array(index)

key found at index

Eg: Linear\_Search

// go through the program

Binary Search

Algorithm:

To perform binary search first array must be sorted.

Step-1: Set the key, starting index (low), ending index (end)

Step-2: while low < end

1. Find the middle index by mid = (start+end)/2
2. If key == mid

key found

break

1. If key <middle

end = mid-1

1. If key > middle

Low = mid+1

Repeat Step-2

Step-3 : If low > end

key is not found .

Eg: Binary\_Search

//go through the program

Eg: Arrays\_Utility\_Class\_Fill\_Method

// go through the program

Eg: Arrays\_Utility\_Class\_Fill\_Method\_Eg2

// go through the program

Eg: Arrays\_Sort\_Method

you cannot use Arrays.sort method in print statement, because all the elements of the array can’t be printed without loop.

System.out.println(Arrays.sort())

you cannot assign Arrays.sort method in to variable and then print it, because all the elements of sorted array cant be stored in single variable .

int x = Arrays.sort()

System.out.println(x)

Eg: Arrays\_Binary\_Search\_Method

if the key is not found at the index, jvm will place the key in suitable place (sorted order) and returns the index with negative value.

Bubble sort / Exchange sort / Sinking sort

Algorithm

Step-1 : Traverse from left to right , compare adjacent elements .

Step-2 : if left side > right side (elements ) swap them

Step-3 : In this way largest element is moved to right most end at first

Step-3 : Repeat Step-2 find 2nd largest element and move to right side end at 2nd position , Do the same to find remaining largest elements until data is sorted.

Eg: Bubble\_Sort

Note: In bubble sort since largest elements are moved to right most end for every iteration , loop should stop before the largest element which is at right most end

So end condition should be ( array.length – outer loop)

In normal array sorting

Smallest element is moved to the left most side end for every iteration , loop should start after the smallest element which is at left most end.

So start condition should be ( outerloop +1)