import java.util.Arrays;

public class ceilingBS {

public static void main(String[] args) {

int[] arr = {2, 3, 5, 9, 14, 16, 18};

*/\*Arrays.sort(arr);*

*System.out.println(Arrays.toString(arr));\*/*

int ans = Ceiling(arr, 0, arr.length-1, 15);

System.out.println(ans);

}

static int Ceiling(int[] arr, int start, int end,int key){

while(start <= end ){

int mid = start + (end -start)/2;

if(key > arr[mid]){

start = mid +1;

}

else if(key < arr[mid]){

end = mid -1;

}

else{

return arr[mid];

}

}

return arr[start];

}

}

All concepts same as original binary search, just return start if target(key isn’t found)

Since, e>s => loop breaks hence print start which is actually ceiling of the target.