Algorithm1

```
public class Algorithm1 {

    public static void main(String[] args) {

        long startTime = System.currentTimeMillis();

        int x = 9999996;

        int n = 10000000;

        int count = 0;

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

            count++;

            if (i == x) {

                System.out.println("We found the values at " + count + " try");

                break;

            }

        }

        long endTime = System.currentTimeMillis();

        System.out.println("Total time :: " + (endTime - startTime) + " ms");

    }

}
```

Algorithm 2

```
public class Algorithm2 {

    public static void main(String[] args) {

        long startTime = System.currentTimeMillis();

        int x = 9999996;

        int n= 10000000;

        int count=0;

        boolean isEven = n%2 == 0;
        if(x>=0){
            if(isEven){
                for (int i = 0; i < n; i=i+2) {
                    count++;
                    if (i == x) {
                        System.out.println("We found the values at "+count+" try");
                        break;
                    }
                }
            }else{
                for (int i = 1; i < n; i=i+2) {
                    count++;
                    if (i == x) {
                        System.out.println("We found the values at "+count+" try");
                        break;
                    }
                }
            }
        }else{
            System.out.println("oops we are expecting vailid value");
        }

        long endTime = System.currentTimeMillis();
        System.out.println("Total time :: "+(endTime-startTime) + " ms");
    }

}
```

Algorithm 3

```
public class Algorithm3 {
    static int count = 0;
    static int findNumber(int[] collection, int start, int end, int x) {
        count++;
        if (end >= start) {
            int mid = start + (end - start) / 2;
            if (collection[mid] == x)
                return mid;
            if (collection[mid] > x)
                return findNumber(collection, start, mid - 1, x);
            return findNumber(collection, mid + 1, end, x);
        }
        return -1;
    }

    public static void main(String[] args) {
        Algorithm3 obj = new Algorithm3();
        int[] numColl = new int[10000000];
        for (int i = 1; i < numColl.length; i++) {
            numColl[i] = i;
        }
        int high = numColl.length - 1;
        long startTime = System.currentTimeMillis();
        obj.findNumber(numColl, 0, high, 9999997);
        System.out.println("We found the values at " + count + " try");
        long endTime = System.currentTimeMillis();
        System.out.println("Total time :: " + (endTime - startTime) + " ms");
    }


}
```