Linear Search Algorithm:

arr = [$\overset{0}{18}$, $\overset{1}{12}$, $\overset{2}{9}$, $\overset{3}{14}$, $\overset{4}{77}$, $\overset{5}{50}$]   Size = 6.

~~i↑~~ ~~i↑~~ ~~i↑~~ i↑✓

Q) Find whether 14 exists in the array or not

If no value found return −1

Time complexity of Linear search

Best · O(1) // constant.

Worst · O(N) $\longrightarrow$ N is size of array

How many checks will the loop make in best case i e element found at 0th index

arr = [8, 9, 12, 18 . . . . . 200 element]

target = 8

1 comparision is best case

arr is now of size 1 Lakhs

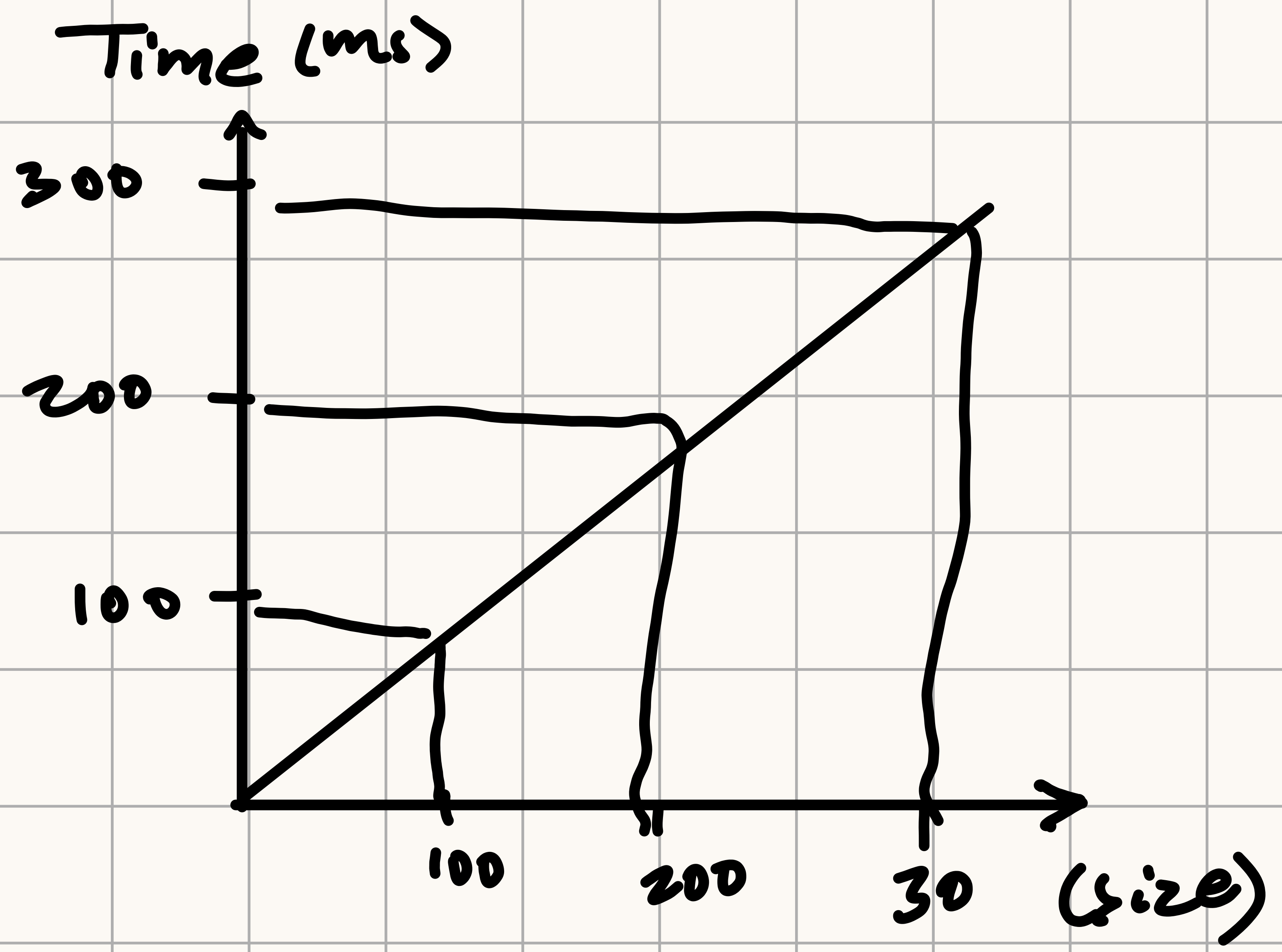
arr = [18, 12, 9, 7, ... 1 lakh item]

target = 18   answer ⇒ only 1 comparision made is the best case

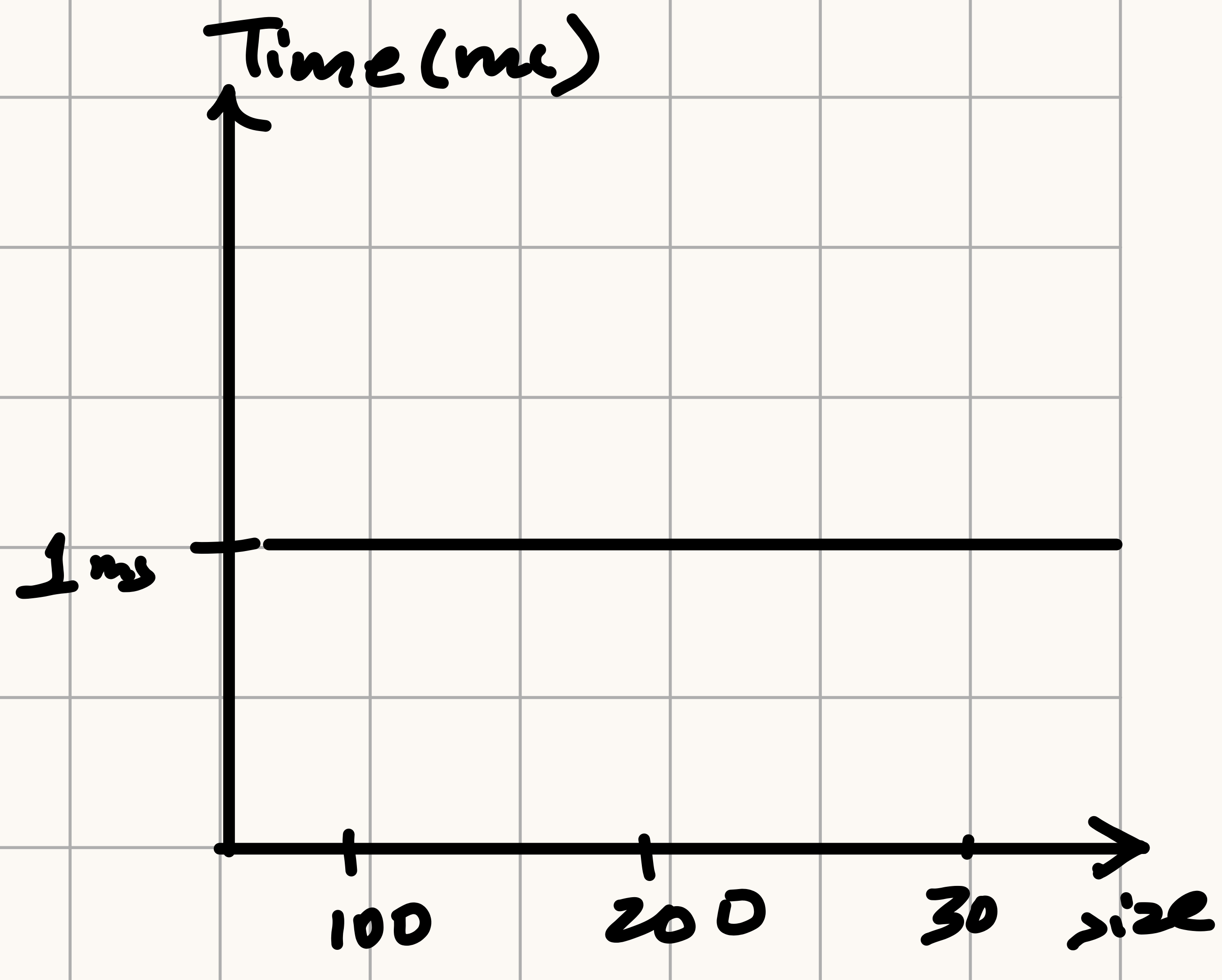
Worst case - You donot find the target item

Iterate / go through every item and then in the end it says I didnot find item

Size of array = 100 → 100 comparision 1ms
1 lakh → 1 lakh comparision 10000 ms
200 → 200 comparision 2ms



Worst case
O(n)



Best case
O(1)

—

Q) Write a Java code to search in the array
Return the index if item found otherwise
item not found return -1.

```
public class main {
        public static void main (String[] args) {
            int[] num = {23, 45, 1, 2, 8, 19, -3, 16};
            int target = 19;
            int ans = linearSearch (nums, target);
            System out println (ans);
        }
// search in the array
                                          input for function      search for
                                          is array                target element
Static int linearSearch (int[] arr, int target) {
        if (arr Length == 0) {
            return -1; // return if doesnot exists
        }
        for (int index = 0; index < arr length; index++) {
            // check for element at every index if it
                                         is equal to target
            int element = arr[index];
            if (element == target) {
                    return index,
            }
        }  return -1; // this line will execute if none
                           of the return statements
                           above have executed.
    }
}
```

Q) Write a Java code to search in the array
Return the element if item found otherwise
item not found return -1.

```
public class Main {
      public static void main (String[] args) {
        int [ ] num {23,45,1, 2,8,19,-3,16};
        int target = 19;
        int ans = Linear Search (nums, target);
        System out println (ans),
    }
  Static int LinearSearch2 (int[] arr, int target)
{  if (arr Length == 0) {
        return -1;
   }
   for (int index = 0, index < arr.Length, index++) {
    int element = arr[index];
    if (element == target) {
          return element;
        }
  }
  return -1;
```

Search in String.

```
public class SearchInStrings {
        public static void main (String [] args) {
            String name = "Kunal";
            char target = 'u';
            System out println (search (name, target)),
        }
    static boolean search (String str, char target) {
            if (str length () == 0) {
                    return false,
             }
             for (int i = 0, i < str length (); i++) {
                    if (target == str charAt (i) ) {
                            return true;
                    }
              }
              return false,
            }
        }
```

Search In Range:

arr = [18, $\boxed{12, -7, 3, 14,}$ 28]

0 1 2 3 4 5.

Q) Search for 3 in the range of index [1, 4]

```
public class SearchInRange {
        public static void main (String[] args) {
                int[] arr = {18, 12, -7, 3.14, 28},
                int target = 3,
                System out println ((linearSearch (arr, target, 1, 4)),
        }
static int linearSearch (int[] arr, int target, int start, int end) {
            if (arr length == 0) {
                return -1,
            }
        for (int index = start; index <= end ; index ++) {
                int element = arr [index];
                if (element == target) {
                        return index,
                }
            }
            return -1;
        }
    }
```

Q) Find minimum number in an array

```
public class FindMin {
        public static void main(String[] args) {
            int[] arr = { 18, 12, -7, 3, 14, 28};
             System out println (min(arr));
        }
        static int min (int[] arr) {
            int ans = arr[0],
            for (int i = 1; i<arr length , i++) {
                if (arr[i]< ans) {
                    ans = arr[i],
                }
            }

            return ans;
        }
```

DSA

+

JAVA

Search in 2D Array:

```
    public class  SearchIn2DArray {
        public static void main (String[] args) {
        int [][] arr = {   // creating 2D array
                    {23, 4, 1},
                    { 18, 12, 3, 9},
                    { 78, 99, 34, 56},
                    { 18, 12}
                 };
                 int target = 34;
  int [] ans = search (arr, target),
  System.out.println (Arrays toString (ans)).

      }
       Static int  search (int[][] arr, int target)
       {
  for (int row=0, row < arr length; row++) {
      for (int col=0, col < arr[row] length;
                                       col++){
            if (arr[row] [col] == target) {
                 return new int[] {row, col},
            }
      }
   } return new int [] {-1,-1};
  }
}
```

Working.

$$arr = \begin{bmatrix} [1, 2, 3] \\ [9, 18, 5] \\ [6, 7, 14] \end{bmatrix}$$

```
for(row=0, row<len;row++);
    {
    for (c=0, c<len(r), c++),
        {if (arr[r][c]==
            target)
          found ans
        }
    }
```

Leetcode Problem 1295.

Q) Find no of nos that have even no. of digits

nums = [18, 124, 9, 1764, 98, 1]

Ans = 3

Method 1.

Convert each number into a string and take the length 1764 ⇒ "1764"

Method 2.

Count the number of digits

1764
176
17
1
0

```
while (n>0) {
    count++,
    n=n/10,
}
```

Count = $\cancel{0}$ $\cancel{1}$ $\cancel{2}$ $\cancel{3}$ $\textcircled{4}$