**Linear Search**

**Implementation:**

int linear\_search (int arr[ ], int n, int a)

{

Int i, index = -1;

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

{

  if ( arr[ ] == a)

{

  index = i;

    break;

 }

}

   return index;

}

**Analysis :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 4 | 3 | 2 | 1 |

So, this is an array with 5 elements that means here, n=5. Now, we want to search the value a=1. At first, we can see that, i=0 of the alliteration which is less than n, so it will enter in the loop.

According to  the condition, if the value is found then it will break and return the index. But in this case the value we want to search is in the last position or last index of  array  which is 4. So, the loop will continue for 5 times or n times here.

**Time complexity (linear search)**

**Best Case:**

If a=5, we can see that , 5 is in the begging of the array. For this, the loop will run for only 1 time. So, the time complexity would be O(1)

**Average Case:**

We know average case = All possible case time / Number of cases

          = (1+2+3+....+n)/n

          =( n(n+1)/2 )/n

          = (n+1)/2;

Ignoring the constant co- efficient, the time  complexity in average case is O(n)

**Worst Case:**

 If there are n elements and the value either exists  in the last position n-1 or not exists ,the loop will run for n times. So, the complexity in worst case of linear search is O(n)