Linear Serach:

Linear search is a very basic and simple search algorithm. In Linear search, we search an element or value in a given array by traversing the array from the starting, till the desired element or value is found.

As we learned in the [previous tutorial](https://www.studytonight.com/data-structures/search-algorithms) that the time complexity of Linear search algorithm is **O(n)**, we will analyse the same and see why it is **O(n)** after implementing it.

**Implementing Linear Search**

Following are the steps of implementation that we will be following:

Traverse the array using a for loop.

1. In every iteration, compare the target value with the current value of the array.
   * If the values match, return the current index of the array.
   * If the values do not match, move on to the next array element.
2. If no match is found, return -1.

To search the number **9** in the array given below, linear search will go step by step in a sequential order starting from the first element in the given array.



// C# code to linearly search x in arr[]. If x

// is present then return its location, otherwise

// return -1

using System;

class GFG

{

public static int search(int[] arr, int x)

{

int n = arr.Length;

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

{

if(arr[i] == x)

return i;

}

return -1;

}

public static void Main()

{

int[] arr = { 2, 3, 4, 10, 40 };

int x = 10;

int result = search(arr, x);

if(result == -1)

Console.WriteLine("Element is not present in array");

else

Console.WriteLine("Element is present at index "+ result);

}

}

# **Data Structure Questions and Answers – Linear Search Iterative**

This set of Data Structure Multiple Choice Questions & Answers (MCQs) focuses on “Linear Search Iterative”.

1. 1. Where is linear searching used?  
   a) When the list has only a few elements  
   b) When performing a single search in an unordered list  
   c) Used all the time  
   d) When the list has only a few elements and When performing a single search in an unordered listView Answer

Answer: d  
Explanation: It is practical to implement linear search in the situations mentioned in When the list has only a few elements and When performing a single search in an unordered list, but for larger elements the complexity becomes larger and it makes sense to sort the list and employ binary search or hashing.

2. Select the code snippet which performs unordered linear search iteratively?  
a)

**int** unorderedLinearSearch(**int** arr[], **int** size, **int** data)

{

**int** index;

**for**(**int** i = 0; i < size; i++)

{

**if**(arr[i] == data)

{

index = i;

**break**;

}

}

**return** index;

}

b)

**int** unorderedLinearSearch(**int** arr[], **int** size, **int** data)

{

**int** index;

**for**(**int** i = 0; i < size; i++)

{

**if**(arr[i] == data)

{

**break**;

}

}

**return** index;

}

c)

**int** unorderedLinearSearch(**int** arr[], **int** size, **int** data)

{

**int** index;

**for**(**int** i = 0; i <= size; i++)

{

**if**(arr[i] == data)

{

index = i;

**break**;

}

}

**return** index;

}

d) None of the mentioned  
View Answer

Answer: a  
Explanation: Unordered term refers to the given array, that is, the elements need not be ordered. To search for an element in such an array, we need to loop through the elements until the desired element is found.

3. What is the best case for linear search?  
a) O(nlogn)  
b) O(logn)  
c) O(n)  
d) O(1)  
View Answer

Answer: d  
Explanation: The element is at the head of the array, hence O(1).

4. What is the worst case for linear search?  
a) O(nlogn)  
b) O(logn)  
c) O(n)  
d) O(1)  
View Answer

Answer: c  
Explanation: Worst case is when the desired element is at the tail of the array or not present at all, in this case you have to traverse till the end of the array, hence the complexity is O(n).

5. Select the code snippet which performs ordered linear search iteratively?  
a)

**public** **int** linearSearch(**int** arr[],**int** key,**int** size)

{

**int** index = -1;

**int** i = 0;

**while**(size > 0)

{

**if**(data[i] == key)

{

index = i;

}

**if**(data[i] > key))

{

index = i;

**break**;

}

i++;

}

**return** index;

}

b)

**public** **int** linearSearch(**int** arr[],**int** key,**int** size)

{

**int** index = -1;

**int** i = 0;

**while**(size > 0)

{

**if**(data[i] == key)

{

index = i;

}

**if**(data[i] > key))

{

**break**;

}

i++;

}

**return** index;

}

c)

**public** **int** linearSearch(**int** arr[],**int** key,**int** size)

{

**int** index = -1;

**int** i = 0;

**while**(size > 0)

{

**if**(data[i] == key)

{

**break**;

}

**if**(data[i] > key))

{

index = i;

}

i++;

}

**return** index;

}

d) None of the mentioned  
View Answer

Answer: b  
Explanation: The term ordered refers to the items in the array being sorted(here we assume ascending order). So traverse through the array until the element, if at any time the value at i exceeds key value, it means the element is not present in the array. This provides a slightly better efficiency than unordered linear search.

advertisement

6. What is the best case and worst case complexity of ordered linear search?  
a) O(nlogn), O(logn)  
b) O(logn), O(nlogn)  
c) O(n), O(1)  
d) O(1), O(n)  
View Answer

Answer: d  
Explanation: Although ordered linear search is better than unordered when the element is not present in the array, the best and worst cases still remain the same, with the key element being found at first position or at last position.

7. Choose the code snippet which uses recursion for linear search.  
a)

**public** **void** linSearch(**int**[] arr, **int** first, **int** last, **int** key)

{

**if**(first == last)

{

System.out.print("-1");

}

**else**

{

**if**(arr[first] == key)

{

System.out.print(first);

}

**else**

{

linSearch(arr, first+1, last, key);

}

}

}

b)

**public** **void** linSearch(**int**[] arr, **int** first, **int** last, **int** key)

{

**if**(first == last)

{

System.out.print("-1");

}

**else**

{

**if**(arr[first] == key)

{

System.out.print(first);

}

**else**

{

linSearch(arr, first+1, last-1, key);

}

}

}

c)

**public** **void** linSearch(**int**[] arr, **int** first, **int** last, **int** key)

{

**if**(first == last)

{

System.out.print("-1");

}

**else**

{

**if**(arr[first] == key)

{

System.out.print(last);

}

**else**

{

linSearch(arr, first+1, last, key);

}

}

}

d)

**public** **void** linSearch(**int**[] arr, **int** first, **int** last, **int** key)

{

**if**(first == last)

{

System.out.print("-1");

}

**else**

{

**if**(arr[first] == key)

{

System.out.print(first);

}

**else**

{

linSearch(arr, first+1, last+1, key);

}

}

}

View Answer

Answer: a  
Explanation: Every time check the key with the array value at first index, if it is not equal then call the function again with an incremented first index.

8. What does the following piece of code do?

**for** (**int** i = 0; i < arr.length-1; i++)

{

**for** (**int** j = i+1; j < arr.length; j++)

{

**if**( (arr[i].equals(arr[j])) && (i != j) )

{

System.out.println(arr[i]);

}

}

}

a) Print the duplicate elements in the array  
b) Print the element with maximum frequency  
c) Print the unique elements in the array  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: The print statement is executed only when the items are equal and their indices are not.

9. Select the code snippet which prints the element with maximum frequency.  
a)

**public** **int** findPopular(**int**[] a)

{

**if** (a == **null** || a.length == 0)

**return** 0;

Arrays.sort(a);

**int** previous = a[0];

**int** popular = a[0];

**int** count = 1;

**int** maxCount = 1;

**for** (**int** i = 1; i < a.length; i++)

{

**if** (a[i] == previous)

count++;

**else**

{

**if** (count > maxCount)

{

popular = a[i-1];

maxCount = count;

}

previous = a[i];

count = 1;

}

}

**return** count > maxCount ? a[a.length-1] : popular;

}

b)

**public** **int** findPopular(**int**[] a)

{

**if** (a == **null** || a.length == 0)

**return** 0;

Arrays.sort(a);

**int** previous = a[0];

**int** popular = a[0];

**int** count = 1;

**int** maxCount = 1;

**for** (**int** i = 1; i < a.length; i++)

{

**if** (a[i] == previous)

count++;

**else**

{

**if** (count > maxCount)

{

popular = a[i];

maxCount = count;

}

previous = a[i];

count = 1;

}

}

**return** count > maxCount ? a[a.length-1] : popular;

}

c)

**public** **int** findPopular(**int**[] a)

{

**if** (a == **null** || a.length == 0)

**return** 0;

Arrays.sort(a);

**int** previous = a[0];

**int** popular = a[0];

**int** count = 1;

**int** maxCount = 1;

**for** (**int** i = 1; i < a.length; i++)

{

**if** (a[i+1] == previous)

count++;

**else**

{

**if** (count > maxCount)

{

popular = a[i-1];

maxCount = count;

}

previous = a[i];

count = 1;

}

}

**return** count > maxCount ? a[a.length-1] : popular;

}

d) None of the mentioned  
View Answer

Answer: a  
Explanation: Traverse through the array and see if it is equal to the previous element, since the array is sorted this method works with a time complexity of O(nlogn), without sorting a Brute force technique must be applied for which the time complexity will be O(n2).

10. Which of the following is a disadvantage of linear search?  
a) Requires more space  
b) Greater time complexities compared to other searching algorithms  
c) Not easy to understand  
d) All of the mentioned  
View Answer

Answer: b  
Explanation: The complexity of linear search as the name suggests is O(n) which is much greater than other searching techniques like binary search(O(logn)).

# **Binary Search**

**Binary Search:** Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.



The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(Log n).

We basically ignore half of the elements just after one comparison.

1. Compare x with the middle element.
2. If x matches with middle element, we return the mid index.
3. Else If x is greater than the mid element, then x can only lie in right half subarray after the mid element. So we recur for right half.
4. Else (x is smaller) recur for the left half.

**Computer Science – Test 7 (Web Design)**  
*MARKS: 50 TIME: 1 hour*

1) Which of the following is not a tag?  
a) <p> b) <pre> c) </a> d) </href>

2) Which of the following tag preserves the format?  
a) <html> b) <pre> c) <body> d) none

3) Find Odd one out.  
a) <table> b) <title> c) <th> d) <tr>

4) Element will contain  
a) Start tag only b) close tag only c) start, close tags and content d) none

5) Which of the following is an example for Element in HTML document  
a) <html> b) <title> c) <html><body> d) <u>welcome</u>

6) Attributes are expressed as  
a) Name=name b) value=name c) name=value d) value=value

7) Which of the following is used to change the background colour of the page into green colour  
a) <body bgcolor=green> b) <color=green>  
c) <body color=green> d) <u>welcome</u>

8) Which of the following is an example of Block level element?  
a) <u> b) <span> c) <p> (d) <title>

9) The face attribute is used to change the  
a) Color of text b) size of text c) name of text d) none

10) Which of the following Heading element will display heading in bigger size  
a) H6 b) H4 c) H1 d) H2

11) Marquee element can be used to  
a) Bold the text b) Scroll the text c) underline the text d) all of them

12) Which is the invalid attribute?  
a) Size b) width c) color d) none

13) OL element is an  
a) Out Line b) Ordered List c) Off Line d) none

14) Which element is used to insert an image in the web page?  
a) IMAGE b) IMG c) SOURCE d) none

15) Which of the following is invalid tag for a table?  
a) <table> b) <row> c) <td> d) <tr>

16) Which tag is used to specify column names/headers for a table?  
a) <tr> b) <title > c) <th> d) <h3>

17) ALIGN attribute of a table is used to align the table to the  
a) left b) right c) left or right d) a) and b)

18) Find the odd one out  
a) <colspan> b) <colgroup > c) <VALIGN> d) <p>

19) Find the odd one out  
a) <colspan> b) <colgroup > c) <VALIGN> d) <p>

20) Border attribute is associated with  
a) <title> b) <table > c) <H1> d) all

21) Which attribute is used to draw border line for a table?  
a) line b) align c) border d) none

22) Which of the following software is used to create a web page using a wizard?  
a) photoshop b) html c) frontpage d) none

23) Which template is used to create a new page?  
a) Site template b) page template c) web template d) none

24) Which object can be used to get inputs from users?  
a) label b) image c) textbox d) all

25) XML stands for  
a) Extending Markup Language b) Extensible Markup Language  
c) Expandable Markup Language d) Extensible Makeup Language

26) Border attribute is associated with  
b) <title> b) <table > c) <H1> d) all

27) Border attribute is associated with  
a) <title> b) <table > c) <H1> d) all

28) In XML, tags are  
a) Case insensitive b) case sensitive c) a) and b) d) none

29) All XML documents start with the declaration of  
a) <XML> b) <? xml version=”1.0” ?> c) <XML version=”1.0”> d) none

30) XML files are saves with the extension  
c) .xml b) .xtml c) .XMLS d) none

31) Which of the following is not true?  
a) XML tags are case sensitive b) XML can have customized tags  
c) XML is used to exchange data between systems d) XML is used to create web pages

32) Identify invalid XML element content  
a) <NO>1</No> b) <name>John</names> c) <10>test<10> d) all

33) Which of the following is true?  
a) XML declaration can appear anywhere b) XML declaration should appear last  
c) XML declaration should appear first in document d) none

34) What is the format of XML element?  
a) <myElement>some content here</myElement> b) <Element content>  
c) <Element>elementname contnet d) none

35) Which object can be used to choose only one of the inputs from users?  
a) Radio button b) list object c) textbox d) none

36) Which is valid statement? In XML  
a) Element is a tag b) tag is user defined c) color attribute is availalbe d) none

37) Which object can be used to choose only one of the inputs from users?  
a) Radio button b) list object c) textbox d) none

38) Which is invalid statement?  
a) XML focuses on data display b) HMTL focuses on data display  
c) a) and b) d) none

39) XML node can contain  
a) Many elements b) only one element c) three elements d) no elements

40) XML is the recommendation by  
a) ASCII b) W3C c) ANSI d) none

41) XHTML is  
a) Extended HTML b) Expanding HTML c) Extensive HTML d) Embedded HTML

42) XHTML is  
a) Extended HTML b) Expanding HTML c) Extensive HTML d) Embedded HTML

43) XML can be converted to other format using  
a) Java b) Databases with XML datatypes c) XML Reader d) all

44) DOCTYPE is mandatory in  
a) HTML b) XML c) XHTML d) none

45) Which is invalid in XHTML?  
a) <p><i>test</p></i> b) <p><i>test</i></p> c) a) and b) d) none

46) XHTML tags are  
b) Case sensitive b) case insensitive c) a) and b) d) none

47) XHTML tags are always  
c) Lower case b) upper case c) a) and b) d) none

48) HTML can be converted to XHTML by including which of the following in first line?  
a) <!DOCTYPE> b) <XHTML> c) a) and b) d) none

49) HTML can be converted to XHTML by  
a) Closing all empty elements b) quoting all attribute values  
c) Changing all element names by lowercase d) all of these

50) HTML can be converted to XHTML by  
b) Closing all empty elements b) quoting all attribute values  
c) Changing all element names by lowercase d) all of these