

[geeksforgeeks.org](https://www.geeksforgeeks.org)

Why is Binary Search preferred over Ternary Search?

3 minutes

The following is a simple recursive **Binary Search** function in C++ taken from [here](#).

```
int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l)
    {
        int mid = l + (r - l)/2;
        if (arr[mid] == x) return mid;
        if (arr[mid] > x) return binarySearch(arr,
l, mid-1, x);
        return binarySearch(arr, mid+1, r, x);
    }
    return -1;
}
```

The following is a simple recursive **Ternary Search** function in C++.

```
int ternarySearch(int arr[], int l, int r, int x)
{
```

```

    if (r >= 1)
    {
        int mid1 = 1 + (r - 1)/3;
        int mid2 = mid1 + (r - 1)/3;
        if (arr[mid1] == x)    return mid1;
        if (arr[mid2] == x)    return mid2;

        if (arr[mid1] > x) return
ternarySearch(arr, l, mid1-1, x);

        if (arr[mid2] < x) return
ternarySearch(arr, mid2+1, r, x);

        return ternarySearch(arr, mid1+1, mid2-1,
x);
    }

    return -1;
}

```

Which of the above two does less comparisons in worst case?

From the first look, it seems the ternary search does less number of comparisons as it makes $\log_3 n$ recursive calls, but binary search makes $\log_2 n$ recursive calls. Let us take a closer look.

The following is recursive formula for counting comparisons in worst case of Binary Search.

$$T(n) = T(n/2) + 2, \quad T(1) = 1$$

The following is recursive formula for counting comparisons in worst case of Ternary Search.

$$T(n) = T(n/3) + 4, \quad T(1) = 1$$

In binary search, there are $2\log_2 n + 1$ comparisons in worst case.

In ternary search, there are $4\log_3 n + 1$ comparisons in worst case.

Time Complexity for Binary search = $2c\log_2 n + O(1)$

Time Complexity for Ternary search = $4c\log_3 n + O(1)$

Therefore, the comparison of Ternary and Binary Searches boils down the comparison of expressions $2\log_3 n$ and $\log_2 n$. The value of $2\log_3 n$ can be written as $(2 / \log_2 3) * \log_2 n$. Since the value of $(2 / \log_2 3)$ is more than one, Ternary Search does more comparisons than Binary Search in worst case.

Exercise:

Why Merge Sort divides input array in two halves, why not in three or more parts?

This article is contributed by **Anmol**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.