Binary Search

- Srivaths P

Binary Search

Binary search is a searching algorithm for a sorted collection of data.

It divides the range to search by half every iteration.

Time complexity: O(logn)

Takes ~20 iterations to search 10^6 elements

Implementation

Finds the last index of target

```
int search(vector<int> a, int target) {
    int left = 0, right = a.size() - 1;

    while (left < right) {
        int mid = (left + right + 1) / 2;

        if (a[mid] <= target) left = mid;
        if (a[mid] > target) right = mid - 1;
    }

    return (a[left] == target) ? left : -1;
}
```

Binary Search Conditions

Binary search works on a set of elements where the “predicate” function applied on it is as follows:

$$T\ T\ T\ \ldots T\ T\ F\ F\ \ldots F\ F\ F$$

Binary search will move:

- L to mid when predicate is true.
- R to mid when predicate is false.

Alternative Binary Search

```
int l = min-1, r = max+1;
while (r-l > 1) {
    int m = (l + r) / 2;
    if (predicate(m))
        l = m;
    else
        r = m;
}

// l is the last true
// r is the first false
```

Interactive Problems:

In interactive problems, you get answers for your queries. Output a query, and an input will be given as the answer.

There will be a limit to the number of queries you can make.

Also note the format of the queries and use it properly.

https://codeforces.com/contest/1480/problem/C

Remove fastio and use endl when solving interactive problems.

Thanks for watching!