ACM Summer Challenge 2020

Searching Algorithms

Searching has always been an important aspect of programming in general. Often, you'll be asked to develop efficient searching algorithms in the corporate world either to provide it as a service, like Google, Microsoft Bing or use it for enhancing the efficiency of your own department.

There are many types of searching algorithms used, but two of them are commonly known to people:

- Linear Search: You visit every element to check whether it's a required element or not. This runs in O(N).
- Binary Search: Usually applied in a sorted list, we first check whether the middle element is less, equal or greater than the required element and then decide our next subtask to get faster to the required element. This runs in O(Log(N)).

For efficient searching, some preprocessing is always required, either it can be sorting the array and making a tree out of a given array (also making sure the tree doesn't get skewed).

With this basic introduction of searching algorithms, here are some links that will guide you further in this domain:
Some links for studying how searching works:
https://www.geeksforgeeks.org/searching-algorithms/
https://codeforces.com/blog/entry/9901

Some more links on how to use searching in C++ using built in functions in <bits/stdc++.h> to save excess coding time:
http://www.cplusplus.com/reference/algorithm/lower_bound/
http://www.cplusplus.com/reference/algorithm/upper_bound/
http://www.cplusplus.com/reference/algorithm/binary_search/

For those of you, who want to try their hands at implementation of searching algorithms and practise using them in questions:
https://codeforces.com/problemset/problem/750/A
https://codeforces.com/problemset/problem/1221/C
https://codeforces.com/problemset/problem/1260/B
https://codeforces.com/problemset/problem/1354/C1
https://codeforces.com/problemset/problem/1249/C2