Advanced Binary Search

Key points about binary search

- Complexity of this algorithm is O(logn)

- It is used whenever there is any monotonic function or we have to search an element.

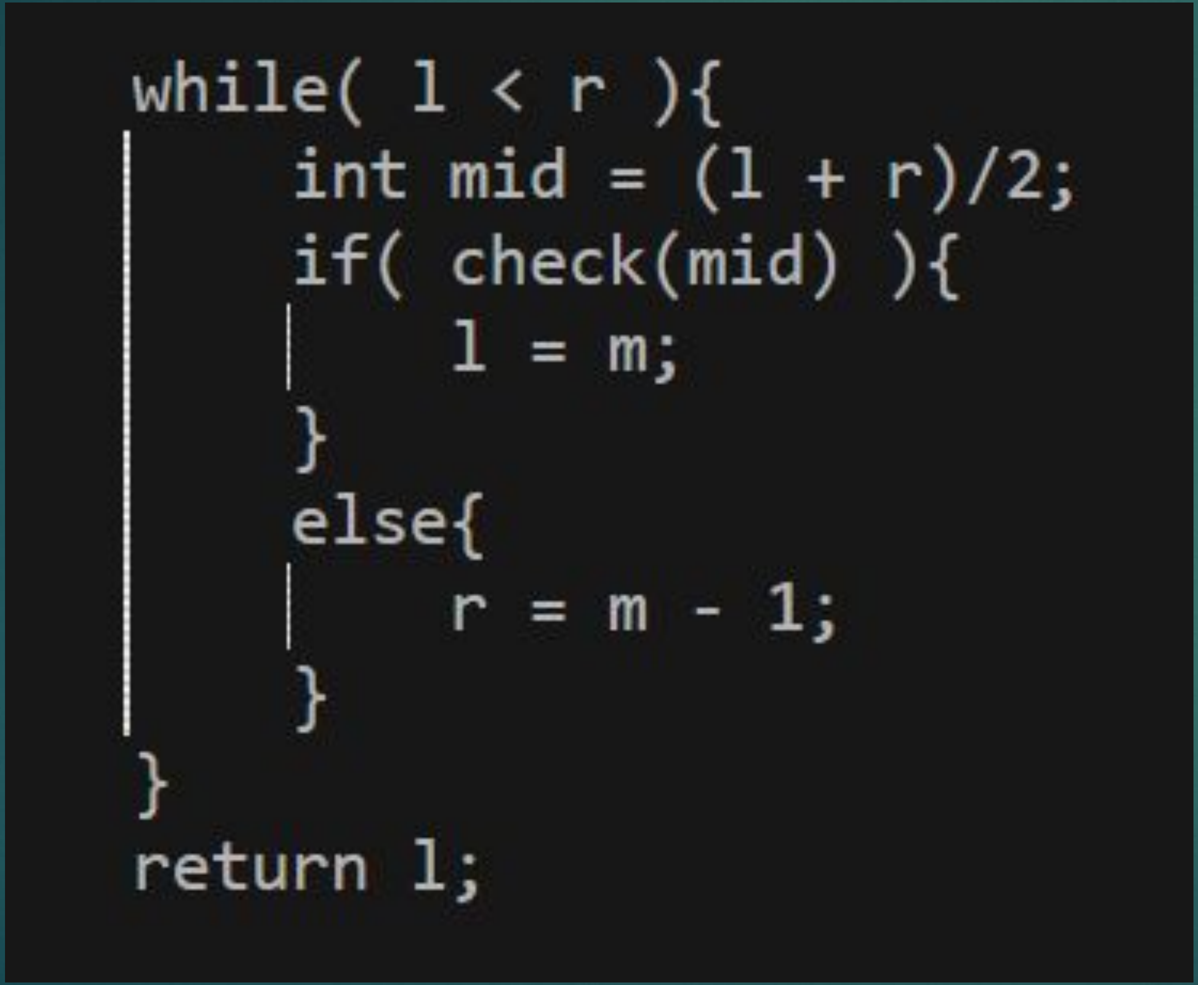
- It is mostly used in sorted lists.

What is a monotonic function?

- A function whose slope is either always negative or always positive, i.e., either it is always increasing or always decreasing.

- f'(x) > 0 (increasing monotonic)
- f'(x) < 0 (decreasing monotonic)
- f(x) > f(y) if x > y (increasing monotonic)
- f(x) < f(y) if x > y (decreasing monotonic)

Basic template for Binary Search



```
while( l < r ){
    int mid = (l + r)/2;
    if( check(mid) ){
        l = m;
    }
    else{
        r = m - 1;
    }
}
return l;
```

Basic Question -

There is a sorted array A. It is shifted to right by some unknown integer k. Given the shifted array, you need to find out the integer k.

$|A| <= 100000$

Example – A = {1,3,5,6,7}

Let k = 3

A transforms to {5,6,7,1,3}

Important points to keep in mind while using binary search

- It might happen a lot of times that binary search end in an infinite loop. This is because sometimes we need to take ceil of mid instead of floor. It varies according to the question.

- Be careful about transitions you make like L=mid+1. Taking the wrong ones may end in an infinite loop.

- Be careful about bounds you take. They might even be overflowing integer range. Also be careful about the initial value L you take. In most cases people start from 1 instead of 0.

Predicate function

- Consider a predicate P defined over some ordered set S (the search space). The search space consists of candidate answers to the problem. In our case, a predicate is a function which returns TRUE or FALSE. We use the predicate to verify if a candidate answer is legal or not.

- Example: We have the set of numbers {1,2,3,4,5}. Our predicate function could be following:

Return TRUE if the number is less than 3 and FALSE otherwise

Now, if we pass 2 to this function, it will return TRUE right?

Basic Problem

Can you find the square root of x in Log(x) time?

Advanced Questions -

- http://www.usaco.org/index.php?page=viewproblem2&cpid=594
- http://www.usaco.org/index.php?page=viewproblem2&cpid=1038

Interactive Problems

What are interactive problems?

Interactive Problems are those problems in which our solution or code interacts with the judge in real time. When we develop a solution for an Interactive Problem then the input data given to our solution may not be predetermined but is built for that problem specifically.

The solution performs a series of exchange of data with the judge and at the end of the conversation the judge decides whether our solution was correct or not.

A simple problem

- In this problem the user has to guess the number during a communication with the judge. The user is provided with the upper and lower bound and he/she can ask the judge whether a number is the number to be guessed. The judge replies with -1 if the number is smaller than the number to be guessed or 1 if number is greater than the number to be guessed or 0 if it is equal to the number to be guessed.

Common errors

- Common error is Idleness Limit Exceeded. This happens when you don't flush the output. Remember to print end of line after outputting an argument and then flush it.

- Wrong answer – There are 2 types of wrong answers here. One is logically incorrect answer. The other is due to printing invalid query. A lot of time it happens that we don't follow the format given and it gives us a wrong answer.

- Another reason of wrong answer is when we output a query which is out of ranges. Like format suggests that it should be in a given range. But the query that you are asking is out of that range so system can't interact with you.

How to check your code for Interactive Problems?

- Many times people don't check their codes and submit directly because they are not able to check on their compilers.

- To solve this problem, you have to act like the system. You have to first make your own answer then check your code. Like for a values already set in system, you see what your code outputs, then think of the input that system should give for this output and give it as input to your code.

- Keep repeating the process till your code doesn’t stop giving output. In this process, you can easily notice exactly what the error is and debugging becomes quite easier this way.

A good problem

- https://codeforces.com/problemset/problem/1520/F1

THANK YOU

By Jaskaran Singh