Beirut Arab University
Faculty of Science
Computer Science Department

Course: CMPS441-Fundamentals of Algorithm
Semester: Fall 2024-2025
Lab1

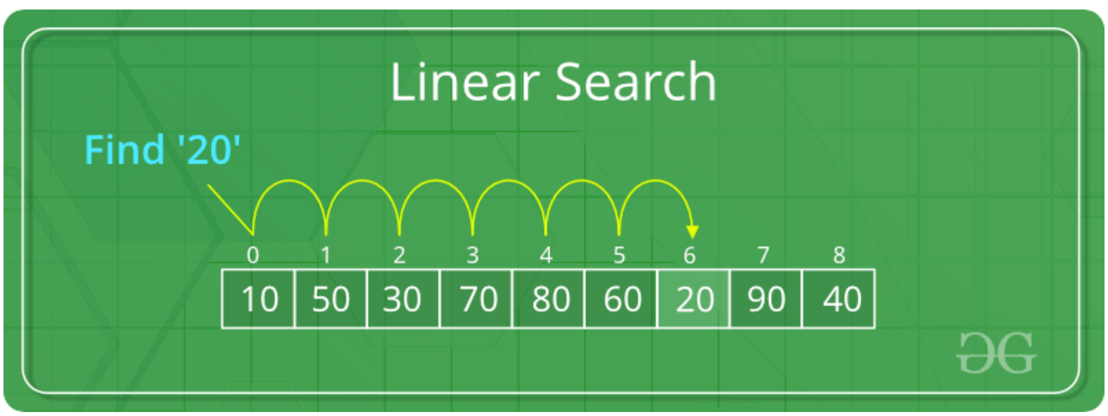
Part I: Linear Search and Binary Search

Given a sorted array arr[] of n elements, write two functions in two different ways to search for a given element x within arr[]. Then, make a comparison between these two approaches.

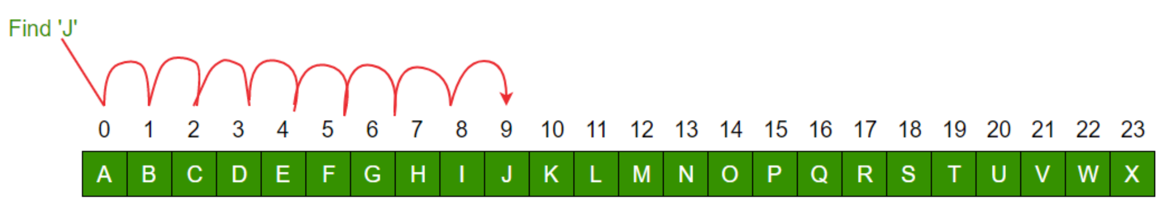
I. A simple approach is to do a **linear search**:

1. Start from the leftmost element of arr[] and compare x with each element of arr[] one by one.
2. If x matches with an element, return the index.
3. If x doesn't match with any of elements, return -1.

The time complexity of the above algorithm is O(n).



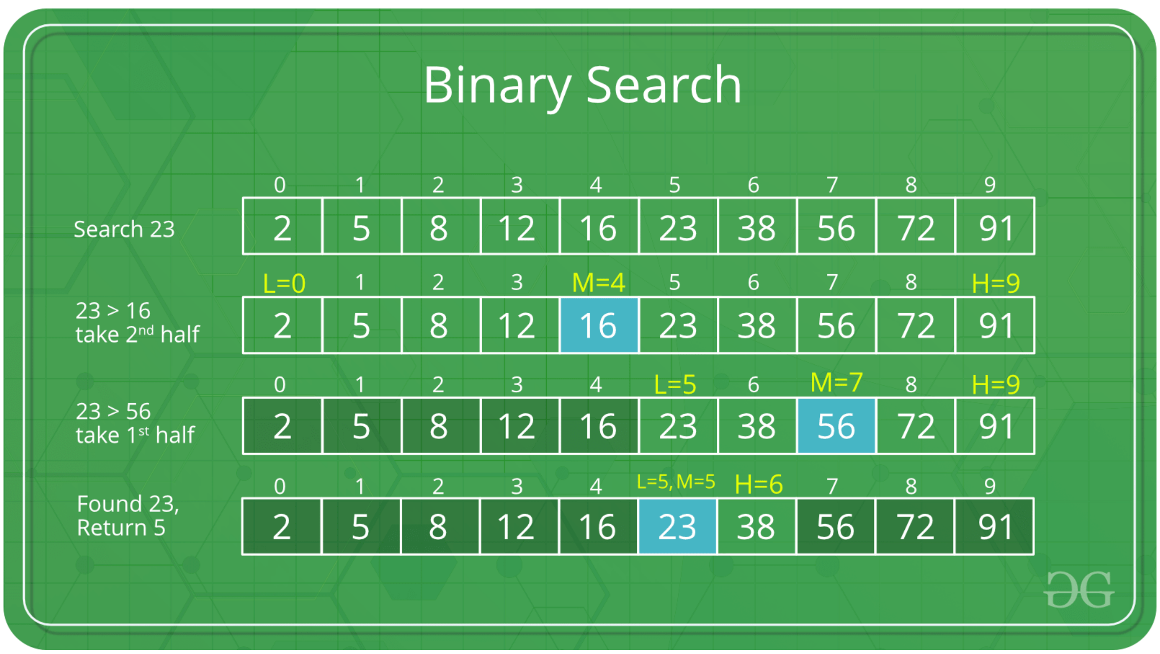
Linear Search to find the element “20” in a given unsorted array



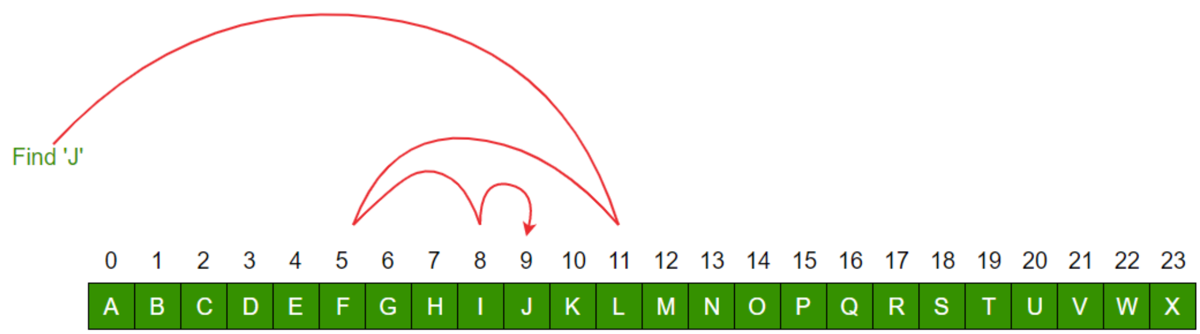
Linear Search to find the element “J” in a given sorted list from A-X

II. **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. Repeat this process until the value is found or the interval becomes empty:

1. Compare x with the middle element.
2. If x matches with the middle element, return the index of the middle element.
3. Else if x is greater than the middle element, then x can only be found in the right half subarray after the middle element. So we recur for the right half.
4. Else if x is smaller than the middle element, recurs for the left half.



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



Binary Search to find the element “J” in a given sorted list from A-X

Part II: A Problem on Binary Search

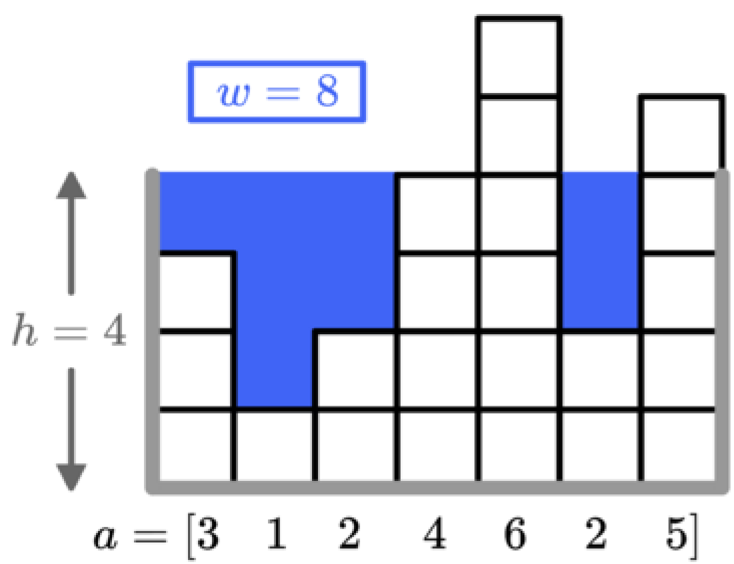
Challenge: Solve the following Codeforces Problem “Building an Aquarium” using Binary Search and aim for an "Accepted" verdict on your submission.
Problem link: https://codeforces.com/problemset/problem/1873/E

Building an Aquarium

You love fish, that's why you have decided to build an aquarium. You have a piece of coral made of n columns, the i-th of which is a_i units tall. Afterwards, you will build a tank around the coral as follows:

- Pick an integer $h \geq 1$ — the *height* of the tank. Build walls of height h on either side of the tank.
- Then, fill the tank up with water so that the height of each column is h, unless the coral is taller than h; then no water should be added to this column.

For example, with $a = [3,1,2,4,6,2,5]$ and a height of $h=4$, you will end up using a total of $w=8$ units of water, as shown.



You can use at most x units of water to fill up the tank, but you want to build the biggest tank possible. What is the largest value of h you can select?

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The first line of each test case contains two positive integers n and x ($1 \le n \le 2 \cdot 10^5$; $1 \le x \le 10^9$) — the number of columns of the coral and the maximum amount of water you can use.

The second line of each test case contains n space-separated integers a_i ($1 \le a_i \le 10^9$) — the heights of the coral.

The sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, output a single positive integer h ($h \ge 1$) — the maximum height the tank can have, so you need at most x units of water to fill up the tank.

We have a proof that under these constraints, such a value of h always exists.

Input

```
5
7 9
3 1 2 4 6 2 5
3 10
1 1 1
4 1
1 4 3 4
6 1984
2 6 5 9 1 8
1 1000000000
1
```

Output

```
4
4
2
335
1000000001
```

Note

The first test case is pictured in the statement. With $h=4$ we need 8 units of water, but if h is increased to 5 we need 13 units of water, which is more than $x=9$. So $h=4$ is optimal.

In the second test case, we can pick $h=4$ and add 3 units to each column, using a total of 9 units of water. It can be shown that this is optimal.

In the third test case, we can pick $h=2$ and use all of our water, so it is optimal.