CSE 4202
Structured Programming II Lab

Lab No: 03
Name of the topic: Binary Search Algorithm and Lower Bound

Fardin Saad
Lecturer, Department of CSE
Talha Ibn Aziz
Lecturer, Department of CSE

October 06, 2020

Contents

1 Tasks

Use Binary Search for all the tasks.

1.1 Task 1:

You will be given the size of a non negative integer array, n and a target variable k in the first line of input (n <= 10^4; k <= 10^4). Then you will be given the array elements in the next line of input as y_1,y_2,...,y_n (1<= y_i<=10^4). Your task is to print "Found" if you find the target variable k in the array otherwise print "Not Found".
Sample Input:
6 3
3 4 1 4 5 2
5 5
1 2 3 1 4
Sample Output:
Found
Not Found

1.2 Task 2:

Write an efficient algorithm that searches for a value k in an m x n matrix. This matrix has the following properties:
1) Integers in each row are sorted from left to right.
2) The first integer of each row is greater than the last integer of the previous row.
Input:
First line of every input contains three integers M, N & k. Next M lines contains N space separated integers as a_1,a_2,...,a_n denoting the elements of the matrix.
Output:
For every input case print "True" if you find k in the m x n matrix else print "False".
Constraints:
1<=n<=m<=k<=1000
1<=a[i]<=1000
Sample Input:
3 4 30
1 3 6 7
10 11 16 19
20 23 30 32
Sample Output:
True
Explanation:
In the sample input 30 which is the target variable is present in the 3rd row and column of the matrix.

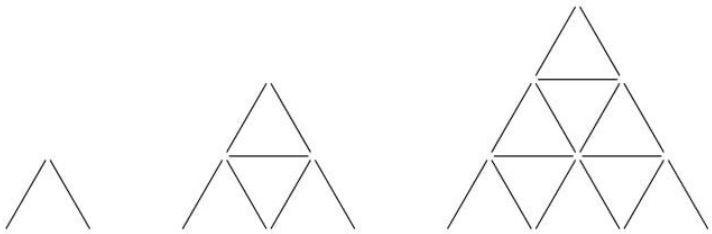
1.3 Task 3:

You are given a sorted array A of size n. Your task is to find the first occurrence and last occurrence of a given number x.
Hint: Don't break your Search after finding the given number, keep on parsing the left and right sub-array until you find first and last occurrence of the given number.

Input:
The first line of input contains the size of the array n and the given number x. Then it is followed by the array elements A_1,A_2,...,A_n .
Output:
Print the first and last occurrence of the given number x. Print "X not Present" if the given number is not present in your array.
Constraints:
1<=n<=10^5
0<=A[i]<=x<=10^5
NOTE: The indexing of the array starts with 0.
Sample Input:
8 5
2 5 5 5 5 6 7 8
Sample Output:
First occurrence of element 5 is at position: 1
Last occurrence of element 5 is at position: 4

1.4 Task 4:

A card pyramid of height 1 is constructed by resting two cards against each other. For h>1, a card pyramid of height h is constructed by placing a card pyramid of height h-1 onto a base. A base consists of h pyramids of height 1, and h-1 cards on top. For example, card pyramids of heights 1, 2, and 3 look as follows:



You start with n cards and **build the tallest pyramid that you can**. If there are some cards remaining, you build the tallest pyramid possible with the **remaining cards**. You repeat this process until it is impossible to build another pyramid. In the end, how many pyramids will you have constructed?
Input:
Each test consists of multiple test cases. The first line contains a single integer t (1<=t<=1000) which is the number of test cases. Next t lines contain descriptions of test cases.
Each test case contains a single integer n (1<=n<=10^9) — the number of cards.
It is guaranteed that the sum of n over all test cases does not exceed 10^9.
Output:
For each test case output a single integer — the number of pyramids you will have constructed in the end
Sample Input:
5
3
14
15
24
1

Sample Output:
1
2
1
3
0
Explanation:
In the first test, you construct a pyramid of height 1 with 2 cards. There is 1 card remaining, which is not enough to build a pyramid.
In the second test, you build two pyramids, each of height 2, with no cards remaining.
In the third test, you build one pyramid of height 3, with no cards remaining.
In the fourth test, you build one pyramid of height 3 with 9 cards remaining. Then you build a pyramid of height 2 with 2 cards remaining. Then you build a final pyramid of height 1 with no cards remaining.
In the fifth test, one card is not enough to build any pyramids.

2 Online Problem Solving

You must solve **5** or more online problems from various platforms such as Codeforces, Hackerrank, CodeChef (–>Clickable links) or any other online platform for problem solving.
N.B: You will be evaluated based on the given tasks and problem solved online in the next lab.