Time: 90min

Q1:

Write a function:

clear Solution { public int solution(int[] A); }

that, given an array A of N integers, returns the smallest positive integer (greater than 0) that does not occur in k.

For example, given A = 11, 3, 6, 4, 1, 2 the function should return 5. Given A = [1, 2, 3], the function should return 4. Given A = -3], the function should return 1.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range 11 100,000),
- each element of array A is an integer within the range [-1,000,000..1,000,000]

Q2:

John has always dreamed of taking a vacation on Hawaii. He has decided to make his dream come true during the next holiday period, and he would like to spend as much time there as possible.

The problem is that there is only one plane per week connecting Hawaii with the city where John lives. The plane departs every Monday and arrives every Sunday. There is no other way to get to Hawaii and back. It means that John can spend only whole weeks in Hawaii.

John knows in which month his vacation starts and in which month it ends. His vacation period starts on the first day of the beginning month and ends on the last day of the ending month. John also knows the year in which his vacation takes place.

For example, if John's vacation lasts from July to September in 2008, then it starts on 1st July 2008 and ends on 30th September 2008.

Your task is to calculate how many weeks John will spend in Hawaii, given:

• the month when the vacation begins;

• the month when the vacation ends;

• the year when the vacation takes place;

• the day of the week for 1st January in the vacation year (for convenience).

The names of the days of the week are: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. The names of the months are: January, February, March, April, May, June, July, August, September, October, November, December. The lengths of the months are, respectively: 31, 28 (or 29 in a leap year), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 days.

pay attention to leap years; you should also consider them. The year of the vacation will be a number between 2001 and 2099, inclusive, and you can tell that the year is a leap year if it's divisible by 4.

write a function:

class Solution { public int solution(int Y, String A, String B, String W); }

that, given an integer Y and three non-empty strings A, B and W, denoting the year of the vacation, the beginning month, the ending month and the day of the week for 1st January of that year, returns the maximum number of weeks hat John can spend in Hawaii.

For example, given Y = 2014, A = "April", B = "May" and W = "Wednesday", the function should return 7, since John can leave his city on April 7th and come back on May 25th, so he will spend 7 weeks on Hawaii (the weeks beginning, respectively, on April 7th, 14th, 21st, 28th and May 5th, 12th, 19th).

Assume that:

- Y is an integer within the range [2,001..2,099];
- A and B are valid month names;
- month B does not preceed month A;
- W is a valid weekday name;
- W is the correct weekday for January 1st of year Yin the actual calendar.

Q3:

You are given an undirected graph consisting of N vertices, numbered from 1 to N, and M edges.

The graph is described by two arrays, A and B, both of length M. A pair (A[K], B[K]), for K from 0 to M-1, describes an edge between vertex A[K] and vertex B[K].

Your task is to check whether the given graph contains a path from vertex 1 to vertex N going through all of the vertices, one by one, in increasing order of their numbers. All connections on the path should be direct.

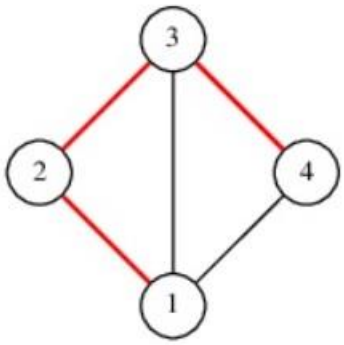
Write a function:

class Solution { public boolean solution(int N, int[] A, int[] B); }

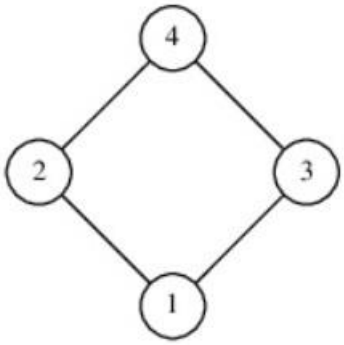
that, given an integer N and two arrays A and B of M integers each, returns true if there exists a path from vertex 1 to N going through all vertices, one by one, in increasing order, or false otherwise.

Examples:

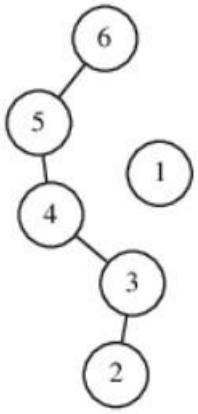
1. Given N = 4, A = [1, 2, 4, 4, 3] and B = [2, 3, 1, 3, 1], the function should return true. There is a path (1 2 3 4) using edges (1, 2), (2, 3) and (4, 3).



2. Given N = 4, A = [1, 2, 1, 3] and B = [2, 4, 3, 4], the function should return false. There is no path (1 2 —> 3 —> 4), as there is no direct connection from vertex 2 to vertex 3.



3. Given N = 6, A = [2, 4, 5, 3] and B = [3, 5, 6, 4], the function should return false. There is no direct connection from vertex 1 to vertex 2.



4. Given N = 3, A = [1, 3] and B = [2, 2], the function should return true. There is a path (1 2 —> 3) using edges (1, 2) and (3, 2).

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [2..100,000];
- M is an integer within the range [0..100,000];
- all elements of arrays A and B are integers within the range [1..N];
- there are no self-loops (edges with A[K] = B[K]) in the graph;
- there are no multiple entries between the same vertices.