**Minimum distance between two numbers: -**

**Easy Accuracy: 19.75% Submissions: 192K+ Points: 2**

You are given an array **a**, of **n** elements. Find the **minimum**index based distance between two distinct elements of the array, **x** and **y**. Return -1, if either **x**or **y**does not exist in the array.

**Example 1:**

**Input:**

N = 4

A[] = {1,2,3,2}

x = 1, y = 2

**Output:** 1

**Explanation:** x = 1 and y = 2. There are

two distances between x and y, which are

1 and 3 out of which the least is 1.

**Example 2:**

**Input:**

N = 7

A[] = {86,39,90,67,84,66,62}

x = 42, y = 12

**Output:** -1

**Explanation:** x = 42 and y = 12. We return

-1 as x and y don't exist in the array.

**Your Task:**  
Complete the function **minDist()**which takes the array **a**, and 3 integers **n, x**and **y**as input parameters and returns the **minimum**distance between **x and y** in the array. Return -1, if either **x**or **y**does not exist in the array.

**Expected Time Complexity:** O(N)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 <= n <= 105  
0 <= a[i], x, y <= 105x != y

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

int minDist(int a[], int n, int x, int y) {

// code here

int xind = INT\_MAX, yind = INT\_MAX;

int mindis = INT\_MAX;

for(int i = 0; i < n; ++i){

if(a[i] == x) xind = i;

else if(a[i] == y) yind = i;

if(xind != INT\_MAX and yind != INT\_MAX)

mindis = min(mindis, abs(xind - yind));

}

if(mindis != INT\_MAX) return mindis;

return -1;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

int a[n];

for (int i = 0; i < n; i++) cin >> a[i];

int x, y;

cin >> x >> y;

Solution obj;

cout << obj.minDist(a, n, x, y) << endl;

}

return 0;

}

// } Driver Code Ends

**T.C: - O(N)**

**S.C: - O(1)**