**Question # 1**

**Fredo and Two Strings <Paysafe>**

Fredo has got two strings ss and tt. Playing around with them, he came up with the following question:  
Will string tt be a subsequence of string ss if he removes substring [si,si+1,....,sj−1,sj][si,si+1,....,sj−1,sj] from string ss ?

Given qq queries, each query consists of indices ii and jj, you have to answer "Yes" if string tt is a subsequence of remaining string ss else answer "No".

**Note**:  
Each query is independent of each other.

**Input Format**:

First line of input consists of string ss.  
Next line consists of string tt.  
Next line consists of two integers qq 22 (qq denoting number of queries and second integer always having value 22).  
Each of following qq lines consists of two integers ii and jj, denoting starting and ending indices of substring to be deleted from string ss.

**Output Format**:  
For each query , output "Yes" if t is a subsequence of remaining string ss, else output "No".  
Answer for each query should come in a new line.

**Input Constraints**:

1≤|s|≤1051≤|s|≤105  
1≤|t|≤|s|1≤|t|≤|s|  
1≤q≤1051≤q≤105  
1≤i≤j≤|s|1≤i≤j≤|s|  
Both strings ss and tt contain lowercase English alphabets.

**Sample Input**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/c439152c91-si.txt?Signature=TCIlTl%2FNNBksaiRReLFwm74k37A%3D&Expires=1510831086&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ" \t "external-link-modal)

abcabcxy

ax

2 2

2 6

6 7

**Sample Output**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/a0a6599479-so.txt?Signature=m%2BJqs32SztwK2oEtylc%2F8iyp718%3D&Expires=1510831087&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ" \t "external-link-modal)

Yes

No

**Explanation**

Query 1: Remaining string s= "axy", clearly "ax" is a subsequence of this string.

Query 2: Remaining string s="abcaby", clearly "ax" is not a subsequence of this string.

**class** SubSequence

{

    // Returns true if str1[] is a subsequence of str2[]

    // m is length of str1 and n is length of str2

**static** **boolean** isSubSequence(String str1, String str2, **int** m, **int** n)

    {

**int** j = 0;

        // Traverse str2 and str1, and compare current character

        // of str2 with first unmatched char of str1, if matched

        // then move ahead in str1

**for** (**int** i=0; i<str1.length; i++)

if(i>m&&j<n)

Continue;

           else  **if** (str1.charAt(j) == str2.charAt(i))

                j++;

        // If all characters of str1 were found in str2

**return** (j==str2.length);

    }

        // Driver program to test methods of graph class

**public** **static** **void** main (String[] args)

    {

        String str1 = "aghjgkgd";

        String str2 = "tyeui";

Scanner sc=new Scanner(System.in);

System.out.println("Enter your first index");

int m=sc.nextInt();

System.out.println("Enter your first index");

int n= sc.nextInt();

**boolean** res = isSubSequence(str1, str2, m, n);

**if**(res)

            System.out.println("Yes");

**else**

            System.out.println("No");

    }

}

//

**Question # 2**

**Common Goods <Paysafe>**

There are NN packets of goods each having some number of items in it.The number of items is in the form of an array AA (**A[i]** items of **ith** type). There are MM number of persons in total each having a share in the goods. They have shares in the form of LL and RR which means that they hold a share of goods **[L....R]**. Bob wants QQ items. He reports the items one by one.  
Each time he takes an item you are required to determine how many people have lost all the items from their share.

**Note**: If Bob wants an item which is unavailable then he will not get it. Here, by "lost all items" we mean that there is not any item left in the share which he holds.

**Input Format**

First line contains an integer NN(number of goods).

Then NN integers follow representing the number of items of each type of good.

Then there is an integer MM(number of persons) followed by the number 22 .

Then we have MM lines containing 22 integers LL and RR representing their share.

Then there is an integer QQ followed by QQ integers (each integer in a separate line) representing goods which Bob wants.

**Output Format**

Print QQ space separated integers representing the number of people who lost all the items of their share.

**Constraints**

1≤N,M,Q≤1051≤N,M,Q≤105

1≤A[i]≤1051≤A[i]≤105

1≤L≤R≤N1≤L≤R≤N

All QQ integers will be in range 11 to NN.

**Sample Input**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/bd159b6297-1samp.txt?Signature=d5nyQZrAwIrv8xckNNmL%2BM%2FToNY%3D&Expires=1510831141&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ" \t "external-link-modal)

4

2 1 1 1

3 2

2 3

1 4

3 3

5

4

3

1

2

1

**Sample Output**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/11202b0898-sout01.txt?Signature=xhixOicC2KNRb6o9esyx5RT%2Fe5A%3D&Expires=1510831141&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ" \t "external-link-modal)

0 1 1 2 3

**Explanation**

After 2nd2nd query goods are 2 1 0 0. So pair **(3,3)** is lost.

After 4th4th query goods are 1 0 0 0. So pairs **(3,3)** and **(2,3)** are lost .

After 5th5th query goods are 0 0 0 0. So all the 33 pairs are lost.

import java.io.\*;

*//import for Scanner and other utility classes*

import java.util.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

int a[]=new int[n+1];

for(int i=1;i<=n;i++)

a[i]=s.nextInt();

int m=s.nextInt();

int t=s.nextInt();

int ar[][]=new int[m][2];

for(int i=0;i<m;i++)

{

ar[i][0]=s.nextInt();

ar[i][1]=s.nextInt();

}

int q=s.nextInt();

int c=0;

while(q-->0)

{

int k=s.nextInt();

if(c<m)

{

a[k]=a[k]-1;

if(a[k]==0)

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

{ int f=0;

if(ar[i][0]!=0||ar[i][1]!=0)

f=1;

if(ar[i][0]==k)

ar[i][0]=0;

if(ar[i][1]==k)

ar[i][1]=0;

if(f==1&&ar[i][0]==0&&ar[i][1]==0)

c++;

}

}

}

System.out.print(c+" ");

}

}

}

**Question # 3**

**Historic Heist <Paysafe>**

Danny Ocean wants to score the biggest heist in history. His target? The MGM Grand. It's not an easy heist so he wants your help.

The city is in the form of a graph, there are NN junctions connected by MM bi-directional roads. The time taken to travel on the ithith road is titi. The MGM Grand is located at the junction with index ss, whereas Danny's safe house is at junction dd. There are several police stations in the city. In the case of any criminal activity, police units are dispatched from all the police stations immediately.   
Now Danny wants to know the least amount of time in which he can reach the safehouse after completing the heist from MGM Grand without being intercepted by the police at any junction of the city.

**Note:** Police can only intercept Danny at any of the junctions in the city.

**Input**

The first line consists of a single integer TT denoting the number of test cases.   
The first line of each test case consists of an integers NN, denoting the number of junctions.   
The second line of each test case contains NN space-separated integers. If the ithith integer is 11, then it means that the ithith has the police station. It is 00 otherwise.   
Next line contains two integers ss and dd, denoting the junction with the MGM Grand and the safehouse respectively.   
Next line of each test case consists of an integer MM and 33, denoting the number of roads in Byteland and the number of columns in the input.   
The following MM lines contain three space-separated integers uu, vv, and tt, where uu and vv are the numbers of the cities connected by this road and tt is the time taken to travel on that road.   
**Note:** Input contains self-loops and multiple roads between two junctions.

**Output**

For each test case, output a single integer denoting the **least amount of time in which Danny can reach his safehouse avoiding the police**. If no such path exists, then print **-1** instead.

**Constraints**

* 1≤T≤101≤T≤10
* 1≤N,M≤1051≤N,M≤105
* 1≤s,d≤N1≤s,d≤N
* 1≤u,v≤N1≤u,v≤N
* 1≤t≤1091≤t≤109

**Sample Input**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/f73c9cd861-Si.txt?Signature=R91RGE80l34yqtD2rbTH3oOBY7E%3D&Expires=1510831202&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ" \t "external-link-modal)

2

4

1 0 0 1

2 3

4 3

1 2 1

1 3 4

2 3 3

3 4 5

2

0 1

2 1

1 3

1 2 1

**Sample Output**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/f79e410461-So.txt?Signature=luSy5v%2BFAqdQ8k9H0xSoNCu2zUc%3D&Expires=1510831202&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ" \t "external-link-modal)

3

-1

**Explanation**

In the first test case, junctions 11 and 44 have police stations.   
MGM Grand is at 22 and safe house at junction 33.   
After committing the crime, Danny can reach the safehouse from junction 22 in 33 units of time without being intercepted by the police at any of the junctions.

In the second test case, junction 22 has a police station.   
MGM Grand is at junction 22 and safehouse at 11.   
Since the police station is at the same junction as MGM Grand police will intercept Danny immediately. Hence, −1−1.

1. import java.util.ArrayList;
2. import java.util.Scanner;

5. class Graph {
6. class Edge {
7. int src, dest, weight;
8. Edge() {
9. src = dest = weight = 0;
10. }
11. };
12. int V, E;
13. Edge edge[];
14. Graph(int v, int e)
15. {
16. V = v;
17. E = e;
18. edge = new Edge[e];
19. for (int i=0; i<e; ++i)
20. edge[i] = new Edge();
21. }
22. public static void main(String[] args) {
23. Scanner sc = new Scanner(System.in);
24. int x = sc.nextInt();
25. for(int i=0;i<x;i++){
26. int N = sc.nextInt();
27. ArrayList<Integer> policeStn = new ArrayList<Integer>();
28. for (int j = 0; j < N; j++) {
29. int input = sc.nextInt();
30. if(input != 0)
31. policeStn.add(j);
32. }
33. int MGMGrand = sc.nextInt() - 1;
34. int safeHouse = sc.nextInt() - 1;
35. int edges = sc.nextInt();
36. sc.nextInt();
37. Graph g = new Graph(N,edges\*2);
38. for (int j = 0; j < edges\*2; j+=2) {
39. g.edge[j].src = sc.nextInt() - 1 ;
40. g.edge[j].dest = sc.nextInt() - 1;
41. g.edge[j].weight = sc.nextInt();
42. g.edge[j+1].src = g.edge[j].dest;
43. g.edge[j+1].dest = g.edge[j].src;
44. g.edge[j+1].weight = g.edge[j].weight;
45. }
46. int[] MGMpaths = shortestPath(g,MGMGrand);
47. int[] PolicePaths = new int[N];
48. int count = 0;
49. for(int j:policeStn){
50. int []dist = shortestPath(g,j);
51. if(count == 0){
52. PolicePaths = dist;
53. }
54. else{
55. for (int k = 0; k < N; k++) {
56. PolicePaths[k] = PolicePaths[k]<dist[k]?PolicePaths[k]:dist[k];
57. }
58. }
59. }
60. if(MGMpaths[safeHouse]>=PolicePaths[safeHouse]){
61. System.out.println(""+-1);
62. }
63. else{
64. System.out.println(""+MGMpaths[safeHouse]);
65. }
66. }
67. }
68. static int[] shortestPath(Graph graph,int src)
69. {
70. int V = graph.V, E = graph.E;
71. int dist[] = new int[V];
72. for (int i=0; i<V; ++i)
73. dist[i] = Integer.MAX\_VALUE;
74. dist[src] = 0;
75. for (int i=1; i<V; ++i)
76. {
77. for (int j=0; j<E; ++j)
78. {
79. int u = graph.edge[j].src;
80. int v = graph.edge[j].dest;
81. int weight = graph.edge[j].weight;
82. if (dist[u]!=Integer.MAX\_VALUE &&
83. dist[u]+weight<dist[v])
84. dist[v]=dist[u]+weight;
85. }
86. }
87. return dist;
88. }
89. }