**liPSG COLLEGE OF TECHNOLOGY, COIMBATORE-4**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**19I510 DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY**

**CA-2**

**CLASS: III B. Tech (IT) G1                                                                            SEM: V**  
**DURATION: 2 Hours                               MAX MARKS: 15**

**Date:29.09.2022**

1. **You have 𝑛 ads to place on a popular Internet page. For each ad, you know how much is the advertiser willing to pay for one click on this ad. You have set up 𝑛 slots on your page and estimated the expected number of clicks per day for each slot. Now, your goal is to distribute the ads among the slots to maximize the total revenue in O(nlogn).**

**Task**

Given two sequences 𝑎1, 𝑎2, …..𝑎𝑛 (𝑎𝑖 is the profit per click of the 𝑖-th ad) and 𝑏1, 𝑏2, ….. 𝑏𝑛 (𝑏𝑖 is the average number of clicks per day of the 𝑖-th slot), we need to partition them into 𝑛 pairs (𝑎𝑖, 𝑏𝑗 ) such that the sum of their products is maximized.

**Input Format**

The first line contains an integer 𝑛, the second one contains a sequence of integers 𝑎1, 𝑎2, ….., 𝑎𝑛, the third one contains a sequence of integers 𝑏1, 𝑏2, …….., 𝑏𝑛.

**Constraints** 1 ≤ 𝑛 ≤ 103; −105 ≤ 𝑎𝑖, 𝑏𝑖 ≤ 105 for all 1 ≤ 𝑖 ≤ 𝑛.

**Output Format**

Output the maximum value of , where 𝑐1, 𝑐2, …… 𝑐𝑛 is a permutation of 𝑏1, 𝑏2, …… 𝑏𝑛.

**Sample Input**

3

1 3 -5

-2 4 1

**Sample Output**

23

1. **You are going to travel to another city that is located 𝑑 miles away from your home city. Your car can travel at most 𝑚 miles on a full tank and you start with a full tank. Along your way, there are gas stations at distances stop1, stop2, . . . , stop𝑛 from your home city. What is the minimum number of refills needed in O(n)?**

**Input Format**The first line contains an integer 𝑑. The second line contains an integer 𝑚. The third line specifies an integer 𝑛. Finally, the last line contains integers stop1, stop2 , . . . ,stopn.

**Output Format**

Assuming that the distance between the cities is 𝑑 miles, a car can travel at most 𝑚 miles on a full tank, and there are gas stations at distances stop1 ,stop2 , . . . ,stop𝑛 along the way, output the minimum number of refills needed. Assume that the car starts with a full tank. If it is not possible to reach the destination, output −1.

**Constraints** 1 ≤ 𝑑 ≤ 1000 . 1 ≤ 𝑚 ≤ 400. 1 ≤ 𝑛 ≤ 300. 0 < stop1 < stop2 < · · · < stop𝑛 < 𝑑

**Sample Input**

950

400

4

200 375 550 750

**Sample Output**

2

1. **Before bridges were common, ferries were used to transport cars across rivers. River ferries, unlike their larger cousins, run on a guide line and are powered by the river’s current. Cars drive onto the ferry from one end, the ferry crosses the river, and the cars exit from the other end of the ferry. There is a ferry across the river that can take n cars across the river in t minutes and return in t minutes. m cars arrive at the ferry terminal by a given schedule. What is the earliest time that all the cars can be transported across the river? What is the minimum number of trips that the operator must make to deliver all cars by that time in O(n)?**

**Input Format**

The first line of input contains c, the number of test cases. Each test case begins with n, t, m. n - cars can be taken across the river; t- minutes and m – no. of cars [m lines follow], each giving the arrival time for a car (in minutes since the beginning of the day). The operator can run the ferry whenever he or she wishes, but can take only the cars that have arrived up to that time.

**Constraints**

0 < n, t, m < 1440 The arrival times for each test case are in non-decreasing order.

**Output Format**

For each test case, output a single line with two integers: the time, in minutes since the beginning of the day, when the last car is delivered to the other side of the river, and the minimum number of trips made by the ferry to carry the cars within that time.

**Sample Input**

2

2 10 10

0

10

20

30

40

50

60

70

80

90

2 10 3

10

30

40

**Sample Output**

100 5

50 2

1. **Given an array of non-negative integers, sort the elements in ascending order using merge sort and print the number of inversions used by the algorithm to indicate how far (or close) the array is from being sorted in Time Complexity: O(n log n).**

**Input Format**

First line contains the number of integers (n)

Next line contains the set of positive integers separated by space.

**Constraint:** integers are always positive 1<=n<=1000

**Output Format**

number of inversions in an array

**Sample Input**

7

1 2 3 4 5 6 8

**Sample Output**

0

**Sample Input**

8

8 4 2 1

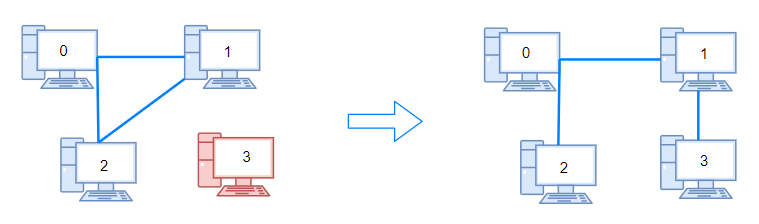
**Sample Output**

6

1. **There are n computers numbered from 0 to n-1 connected by ethernet cables connections forming a network where connections[i] = [ai, bi] represents a connection between computers ai and bi. Any computer can reach any other computer directly or indirectly through the network.**

**You are given an initial computer network connection. You can extract certain cables between two directly connected computers, and place them between any pair of disconnected computers to make them directly connected in O(n+m) where m is the length of connections and n is the number of nodes.**

**Return the minimum number of times you need to do this in order to make all the computers connected. If it is not possible, return -1.**

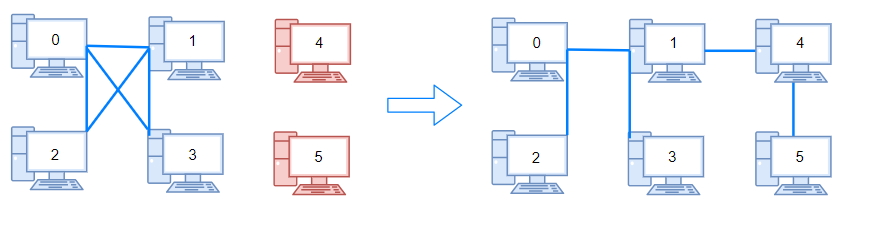
 **Example 1:**

Input: n = 4, connections = [[0,1],[0,2],[1,2]]

Output: 1

Explanation: Remove cable between computer 1 and 2 and place between computers 1 and 3.

**Example 2:**



Input: n = 6, connections = [[0,1],[0,2],[0,3],[1,2],[1,3]]

Output: 2

**Example 3:**

Input: n = 6, connections = [[0,1],[0,2],[0,3],[1,2]]

Output: -1

Explanation: There are not enough cables.

**Constraints:**

* 1 <= n <= 105
* 1 <= connections.length <= min(n \* (n - 1) / 2)
* connections[i].length == 2 (bidirection)
* 0 <= ai, bi < n
* ai != bi
* There are no repeated connections.
* No two computers are connected by more than one cable.

1. **There are n cities and m flight connections. Your task is to check if you can travel from any city to any other city using the available flights in O(V+E).**

**Input Format**  
The first input line has two integers n and m denoting the number of cities and flights. The cities are numbered 1,2,…,n.

After this, there are m lines describing the flights. Each line has two integers a and b indicating that there is a flight from city a to city b. All flights are one-way flights.  
**Output Format**  
Print "YES" if all routes are possible, and "NO" otherwise.

**Sample Input**

4 5  
1 2  
2 3  
3 1  
1 4  
3 4  
**Sample Output**

NO