Indian Institute of technology, Guwahati
Department of Computer Science and Engineering
Data Structure Lab: (CS210)

Offline Assignment: 4

Date: 28th August 2017. **Total Marks: 20**
Deadline: 10PM, 3rd September 2017. (Hard Deadline)

1. **[Sort-Reduce]** You have probably heard about Map-Reduce, but it is different. In Sort-Reduce technique, you will be given an integer array and in each step, you sort it and reduce it by taking absolute differences of adjacent elements. You can have negative numbers as well in the initial array.

   In more details, initially you are given an integer array and you repeat following operations until array has a single element:

   Sort: sort the current array A in ascending order.

   Reduce: Conceptually create a new array B by taking absolute differences of every two adjacent elements of A. Thus the length of B will be one less than the length of A. The array B will be the current array for next iteration. You can actually perform reduce in array A itself. Make you algorithm step **in-place, i.e.** you should not use array B.

   Finally, when you get array with a single element, print it. **[10]**

   Constraint:
   1<= Length of initial array <= 10^6
   10^-9 <= Element of initial array <= 10^9

   Input:
   First line of input will be N, the length of initial array.
   Next line will have N space separated integers which are elements of A.

   Output:
   In a single line, print the required output as explained in problem statement.

   Test 1:

   Input:
   5
   1 17 2 15 8

   Output:
   3

   Explanation:
   Sort initial array: 1, 2, 8, 15, 17. Reduce: 1, 6, 7, 2.
   Sort: 1, 2, 6, 7. Reduce: 1, 4, 1.

Sort: 1, 1, 4. Reduce: 0, 3.
Sort: 0, 3. Reduce: 3

Test 2:
Input:
6
1 -1 4 2 11 -5

Output:
0

Explanation:
Sort initial array: -5,-1,1,2,4,11. Reduce: 4,2,1,2,7.
Sort: 1, 2, 2, 4, 7. Reduce: 1, 0, 2, 3
Sort: 0, 1, 2, 3. Reduce: 1, 1, 1.
Sort: 1, 1, 1. Reduce: 0, 0
Sort: 0, 0. Reduce: 0

2. **[Maximum Fare]** There are N cities in our country. Some of them are connected by road. You are given R records of '**Road Information**'. Each record from road information will be in the form of <f, s, d>, where d is the distance in kilometers between cities f and s which are directly connected by road. Note that f and s are different. If there is road between f and s means that there is a road between s and f. You can assume that the cities for which road information is not given, are not connected by road.

   You are also given F records of '**Fare Information**' for some cities. Each record from fare information is of the form <f, s, p>, where p is per kilometer fare in rupees between cities f and s which are directly connected by road. Note that fare information between s and f will be same as fare information between f and s.

   Now being a travel enthusiast, you want to know maximum total fare between two cities which are directly connected by road. During calculation, you can ignore the cities for which road information or fare information is missing.
   Print the names of two cities which have maximum fare and also print the fare between them. In case, cities f and s have the maximum fare and also cities f' and s' have the same fare, you can choose any pair either f and s, or f' and s' for your answer.
   Consider both Store Road Information and Fare information are **sparse matrices**. **Store Road Information as array and Fare information as linked list.** **[10]**

   Constraint:
   1<= N, R, F, d, p <= 10^6
   Name of a city will be a string of length at most 20.

   Input:
   First line of input will be space separated integers N, R, F.
   For next R lines, each is record of road information i.e. space separated f, s and d.
   For next F lines, each is record of fare information i.e. space separated f, s and p.

Output:
In a single line, print space separated names of the two cities with maximum fare and the fare.

Test 1:

Input:
5 5 4
Delhi Mumbai 25
Delhi Indore 98
Mumbai Kolkata 12
Kolkata Delhi 10
Indore Pune 71
Delhi Mumbai 15
Mumbai Kolkata 13
Delhi Kolkata 21
Indore Kolkata 12

Output:
Delhi Mumbai 375

Explanation:
Delhi and Mumbai are directly connected by road. Distance between Delhi and Mumbai is 25 kilometers and per kilometer fare is 15. So fare between Delhi and Mumbai is 25*15=375 which more than the fare between any other two cities. Note that the answer “Mumbai Delhi 375” is also correct.

Test 2:

Input:
6 5 5
Guwahati Shillong 50
Mumbai Pune 20
Mumbai Agra 120
Agra Kanpur 80
Kanpur Mumbai 100
Pune Mumbai 30
Guwahati Kanpur 80
Guwahati Shillong 19
Kanpur Agra 10
Kanpur Mumbai 40

Output:
Kanpur Mumbai 4000