

# **Direct Connections**

**-Crack a hack**

**USN: 01FE15BEC045**

**NAME: ASHREET N SALIMATH**

## Problem statement:

Enter-View(EV) is a linear, street-like country. By linear, we mean all the cities of the country are placed on a single straight line - the  $x$ -axis. Thus every city's position can be defined by a single coordinate,  $x_i$  the distance from the left borderline of the country.

You can treat all cities as single points.

Unfortunately, the dictator of telecommunication of EV (Mr. S. Treat Jr.) doesn't know anything about the modern telecom technologies, except for peer-to-peer connections. Even worse, his thoughts on peer-to-peer connections are extremely faulty: he believes that, if  $P_i$  people are living in city  $i$ , there must be at least  $P_i$  cables from city  $i$  to every other city of EV - this way he can guarantee no congestion will ever occur!

Mr. Treat hires you to find out how much cable they need to implement this telecommunication system, given the coordination of the cities and their respective population.

### Input Format

A number  $T$  is given in the first line and then comes  $T$  blocks, each representing a scenario.

Each scenario consists of three lines. The first line indicates the number of cities ( $N$ ). The second line indicates the coordinates of the  $N$  cities. The third line contains the population of each of the cities. The cities needn't be in increasing order in the input.

### Output Format

For each scenario of the input, write the length of cable needed in a single line modulo  $1,000,000,7$ .

### Sample Input

```
2
3
1 3 6
10 20 30
5
5 55 555 55555 555555
3333 333 333 33 35
```

### Sample Output

```
280
463055586
```

## Explanation:

For the first test case, having 3 cities requires 3 sets of cable connections. Between city 1 and 2, which has a population of 10 and 20, respectively, Mr. Treat believes at least 10 cables should come out of city 1 for this connection, and at least 20 cables should come out of city 2 for this connection. Thus, the connection between city 1 and city 2 will require 20 cables, each crossing a distance of  $3-1=2$  km. Applying this absurd logic to connection 2,3 and 1,3, we have

[1,2] => 20 connections X 2km=40 km of cable

[2,3] => 30 connections X 3k=90 km of cable

[1,3] => 30 connections X 5km=150 km of cable

For a total of 280, Output is 280 km of cable

## Solution:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    long long int t,n,distance,contion,n_cables;
```

```
    scanf("%lld",&t);
```

```
    long long int cable[t];
```

```
    for(int j=0;j<t;j++)
```

```
    {
```

```
        cable[j]=0;

    }

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

    {

        scanf("%lld",&n);

        long long int city[n],person[n];

        for(long long int j=0;j<n;j++)

        {

            scanf("%lld",&city[j]);

        }

        for(long long int j=0;j<n;j++)

        {

            scanf("%lld",&person[j]);

        }

        for(long long int j=0;j<n-1;j++)

        {

            for(long long int k=j+1;k<n;k++)

            {

                if(person[k]>person[j])
```

```

        contion=person[k];

    else

        contion=person[j];

        distance=abs(city[k]-city[j]);

        n_cables = contion*distance;

        cable[i]=cable[i] +n_cables;

    }

}

}

for(long long int i=0;i<t;i++)

    printf("%lld\n",cable[i]%1000000007);

    return 0;

}

```

## References:

- 1) <https://www.geeksforgeeks.org/>
- 2) <https://www.cs.umd.edu>

