**GUVI – PROBLEM SETTER**

**SHORTEST REACH**

**PROBLEM STATEMENT:**

Ram is a fruit seller; he wants to travel along different cities in Coding Land and sell his fruits. For moving from one city to another, there includes a traveling cost. There are many such routes. Ram wishes to travel all the cities in Coding Land in such a way that the total cost of travelling from the source city to the destination city is minimum.

Also, Ram decides to sell the fruit with minimum cost in the first city where he visits, and the fruit with second minimum cost in the second city and so on. Ram has 7 different varieties of fruits, each with different cost.

**For** **example**, if he has 7 fruits with prices (87, 34, 90, 99,60, 12,45) for each, he decided to sell,

Suppose, if there are 7 cities to visit,

1st city – fruit with cost 12

2nd city – fruit with cost 34

3rd city – fruit with cost 45 and so on.

Suppose, if there are 9 cities to visit,

1st city – fruit with cost 12

2nd city – fruit with cost 34 and so on till for 7th city.

8th  city – fruit with cost 12

9th city – fruit with cost 34.

Given an unsorted array which has the cost of various fruits that Ram wants to sell.

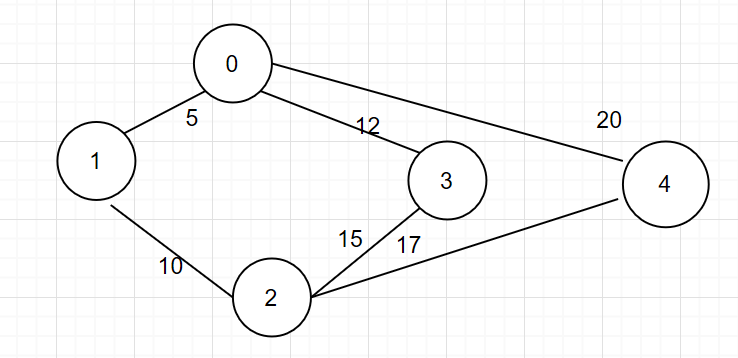
As the greatest programmer of Coding Land, Ram wants you to help him so that he can move and sell his fruits efficiently.

Also, after finding the efficient route, given a order of a city, find the cost of the fruit which Ram decides to sell in that particular city.

From the above example, if given order is 9, the cost of the fruit which ram sells in 9th city (i.e.) 34 should be displayed.

Coding Land has ‘n’ cities numbered from 0 to n. The cities are connected by ‘e’ bidirectional roads.

The following figure is a sample map of Coding Land, the circles denote the cities and the lines denote the routes connecting them. The numbers above the lines are the cost in dollars for travelling from one city to another.



**INPUT FORMAT:**

The first line contains a single integer ‘fr’, that represents the number of fruits.

This is followed by ‘fr’ integer separated by space, each representing the cost of fruit.

The next line contains a single integer, that denotes the number of cities.

The next line contains the number of routes ‘e’. This is followed by e rows, each row contains 3 integers separated by space (e1, e2, c).

The first and second integer represents the source and the destination cities. The third integer represent the cost of travelling from city e1 to city e2.

Last line of the input (i.e.) search, represents the city order or position.

**CONSTRAINTS:**

1<=fr<=100

1 <= n <= 100

1<= e <= 1000

1<= search <=number of cities.

Each route connects two cities.

**OUTPUT FORMAT:**

Print an integer that denotes the minimum cost of travelling all the cities in Coding Land.

Followed by the route.

On the next line, print the cost of the fruit that Ram sells in search city.

**EXAMPLE:**

**SAMPLE INPUT (for above program)**

Number of fruits:7

89 23 65 12 90 45 10

Number of cities: 5

Number of routes: 6

Travelling Cost from city 1 to city 2:

1 0 5

1 2 10

0 3 12

2 3 15

2 4 17

0 4 20

City position: 5

**SAMPLE OUTPUT:**

PATH

1 🡨🡪 0

2 🡨🡪 1

3 🡨🡪 0

4 🡨🡪2

Efficient Travelling Cost = 44

Cost of the fruit sold in 5th city: 65

From the graph, the cost of travelling from city 0 to city 4 is minimum, when Ram visits the city in the following order:

Visits city 0 from city 1: Cost = 5

Visits city 2 from city 1: Cost = 5+10

Visits city 3 from city 0: Cost = 5+10+12

Visits city 4 from city 2: Cost=5+10+12+17 = 44

Since the declared array is {89,23,65,12,90,45,10},

The cost of the fruit sold in 1st City: 10

The cost of the fruit sold in 2nd City: 12

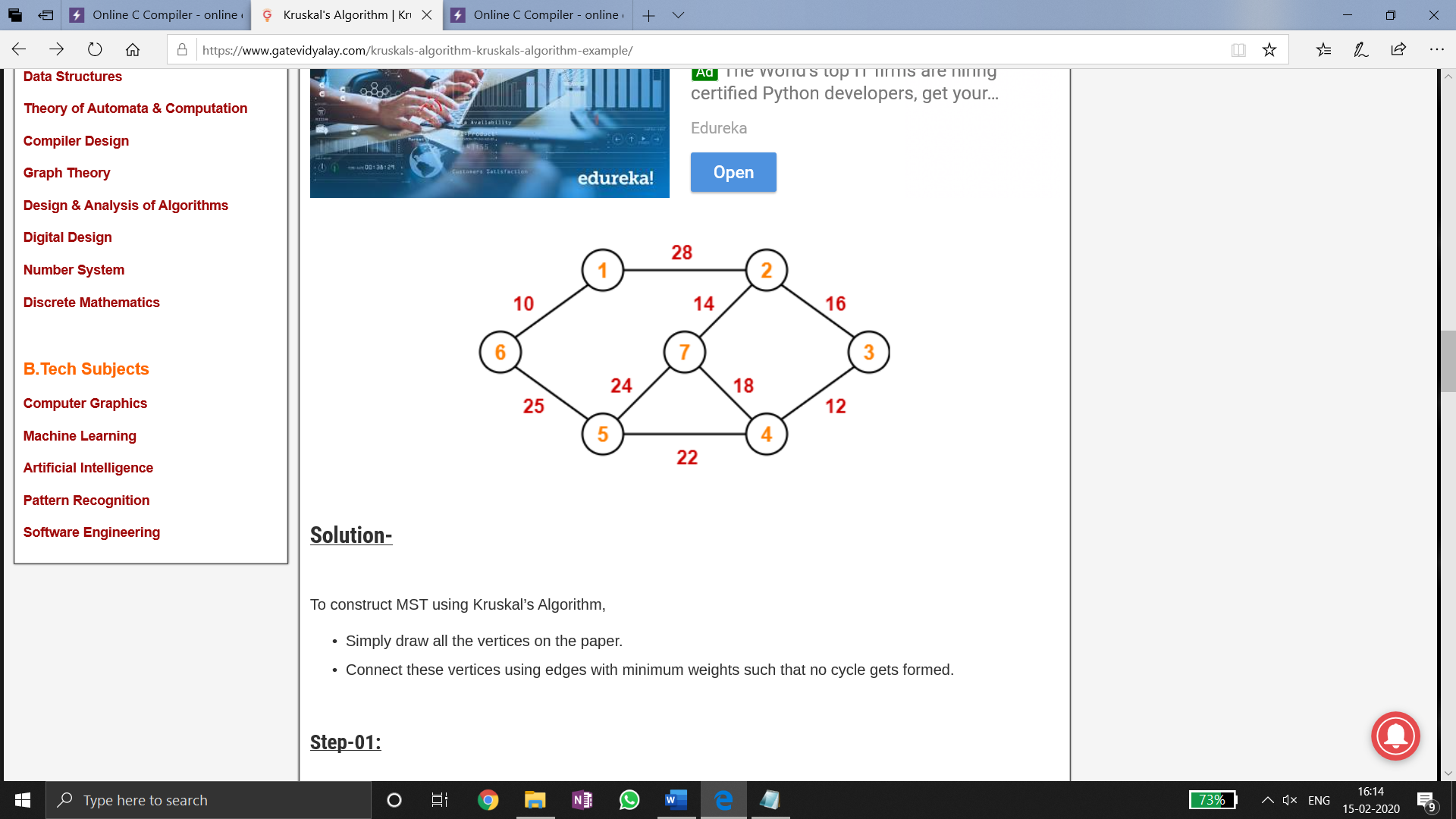
The Cost of the fruit sold in 3rd City: 23

The Cost of the fruit sold in 4th City: 45

The Cost of the fruit sold in 5th City: 65

**TEST CASES:**

**TEST CASE :1**



**INPUT:**

Number of fruits: 5

45 32 78 76 55

Number of cities: 7

Number of routes: 9

Travelling Cost from source to destination city:

5 0 10

5 4 25

4 6 24

0 1 28

6 1 14

4 3 22

6 3 18

3 2 12

1 2 16

Enter the city position: 7

**OUTPUT**:

Path

5 🡨🡪 0

3 🡨🡪 2

6 🡨🡪 1

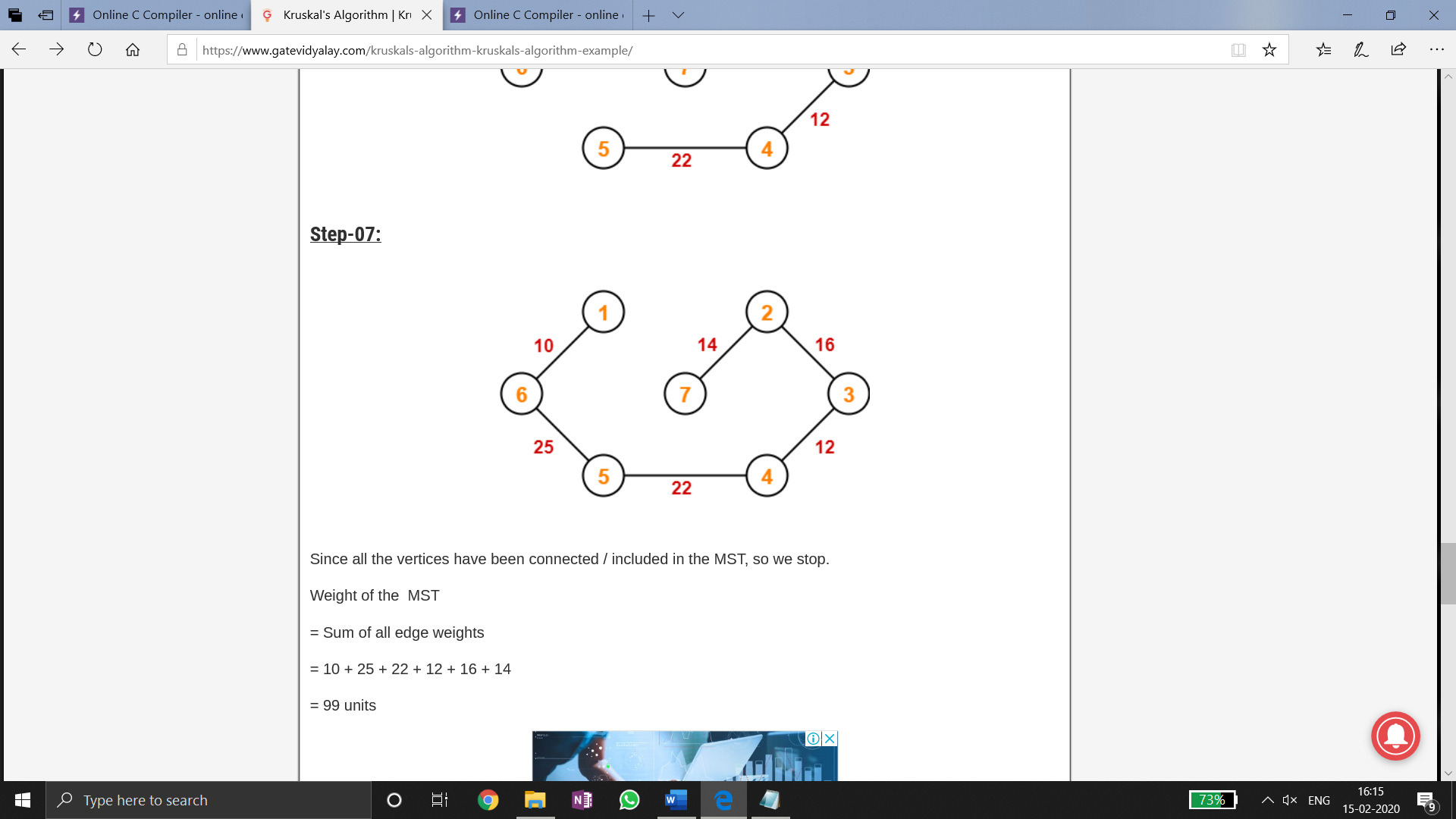
2 🡨🡪1

4 🡨🡪 3

5 🡨🡪 4

Efficient Travelling Cost = 99

Cost of the fruit sold in 7th city: 45



**TEST CASE: 2**

**INPUT**:

Number of fruits: 4

76 56 88 46

Number of cities: 5

Number of routes: 7

Travelling Cost from source to destination city:

0 1 1

0 2 7

1 2 3

0 3 10

2 3 4

0 4 5

4 3 2

Enter the city position: 4

**OUTPUT:**

PATH:

1 🡨🡪 0

4 🡨🡪3

2 🡨🡪 1

3 🡨🡪2

Efficient Travelling Cost: 10

Cost od the fruit sold in 4th city: 88

**TEST CASE: 3**

**INPUT:**

Number of fruits: 4

76 56 88 46

Number of cities: 5

Number of routes: 6

Travelling Cost from source to destination city:

0 1 2

0 2 1

0 3 2

3 2 2

1 2 1

3 4 5

Enter the city position: 7

**OUTPUT**:

Invalid Input. Given City position is exceeds the number of cities in Coding Land.

**TEST CASE: 4**

Number of fruits: 6

76 56 88 46 43 69

Number of cities: 6

Number of routes: 9

Travelling Cost from source to destination city:

0 1 7

0 5 8

1 5 3

1 2 6

5 2 4

2 4 2

5 4 3

2 3 5

4 3 2

Enter the city position: 6

**OUTPUT:**

PATH

4 🡨🡪 2

4 🡨🡪 3

5 🡨🡪 1

5 🡨🡪4

1 🡨🡪 0

Efficient Travelling Cost: 17

Cost of the fruit sold in 6th city:88

**TEST CASE: 5**

**INPUT**:

Number of fruits: 4

76 56 88 46

Number of cities: 5

Number of routes: -6

**OUTPUT:**

INVALID INPUT.

**METHOD USED TO SOLVE:**

**Krushal’s** **Alogorithm** : To find the cost of mininum spanning tree

**Binary Search Tree** : To sort the cost of fruits and traverse it using in-order traversal.