

33m left

**BETA** Can't read the text? [Switch](#) theme

## 41. Cheap Airfare

ALL

### Problem Statement

Anjali has started a new travel agency. She noticed that the air fares between the cities are not symmetrical. That is, the cost of flying from a to b is not the same as that of b to a. Further, there is no guarantee that once you fly directly from a to b there is a direct return flight.

She knows that passengers do not mind changing flights as long as the total airfare is minimized. She needs to advertise the lowest round trip cost between any pair of cities of tourist interest to attract potential passengers. All valid flights i.e. (A, B, P(A,B)) where A is the source city of tourist interest, B is the destination city of tourist interest and P(A,B) is the price of the ticket.

You must help her determine the lowest round trip cost & the cities involved.

### Input Format

The first line consists of integer n indicating the number of cities

The second line consists of integer m indicating the number of flights

Each of the next 'm' lines contains 3 integers separated by a space - starting city A, end city B and the price of the ticket P(A,B) for flight from A to B.

Language

C



Autocomplete Ready



```
20 |
21 | /*
22 |  * Complete the 'CheapestRoundTrip' function below.
23 |  *
24 |  * The function is expected to return an INTEGER_ARRAY.
25 |  * The function accepts following parameters:
26 |  * 1. INTEGER NumCities
27 |  * 2. 2D_INTEGER_ARRAY FlightInfo
28 |  */
29 |
30 | /*
31 |  * To return the integer array from the function, you should:
32 |  * - Store the size of the array to be returned in the result_count variable
33 |  * - Allocate the array statically or dynamically
34 |  *
35 |  * For example,
36 |  * int* return_integer_array_using_static_allocation(int* result_count) {
37 |  *     *result_count = 5;
38 |  *
39 |  *     static int a[5] = {1, 2, 3, 4, 5};
40 |  *
41 |  *     return a;
42 |  * }
43 |  *
44 |  * int* return_integer_array_using_dynamic_allocation(int* result_count) {
45 |  *     *result_count = 5;
46 |  *
47 |  *     int *a = malloc(5 * sizeof(int));
48 |  *
49 |  *     for (int i = 0; i < *result_count; i++) {
```

Line: 20 Col: 1

Test Results

Custom Input

Run Code

Run Tests

Submit

33m left

**Constraints**

1 < Number of cities < 100  
0 < Number of flights < 5000  
0 < Price < 10000

**Output Format**

Return an array of 3 integers. 1st integer is the price for the cheapest round trip. 2nd and 3rd integers are the city indices involved in increasing order.

If there is no round trip available, return 1000000, -1 and -1.

If there are multiple pair of cities with lowest round trip fair, please select the pair with the lowest sum of their indices.

**Sample Input 0**

```
4
6
0 1 20
1 0 20
1 2 10
2 0 10
1 3 30
3 2 30
```

**Sample Output 0**

```
40
0
1
```

**Explanation 0**

Language

C



Autocomplete Ready



```
23  *
24  * The function is expected to return an INTEGER_ARRAY.
25  * The function accepts following parameters:
26  * 1. INTEGER NumCities
27  * 2. 2D_INTEGER_ARRAY FlightInfo
28  */
29
30 /*
31  * To return the integer array from the function, you should:
32  * - Store the size of the array to be returned in the result_count variable
33  * - Allocate the array statically or dynamically
34  *
35  * For example,
36  * int* return_integer_array_using_static_allocation(int* result_count) {
37  *     *result_count = 5;
38  *
39  *     static int a[5] = {1, 2, 3, 4, 5};
40  *
41  *     return a;
42  * }
43  *
44  * int* return_integer_array_using_dynamic_allocation(int* result_count) {
45  *     *result_count = 5;
46  *
47  *     int *a = malloc(5 * sizeof(int));
48  *
49  *     for (int i = 0; i < 5; i++) {
50  *         *(a + i) = i + 1;
51  *     }
52  * }
```

Line: 20 Col: 1

Test Results

Custom Input

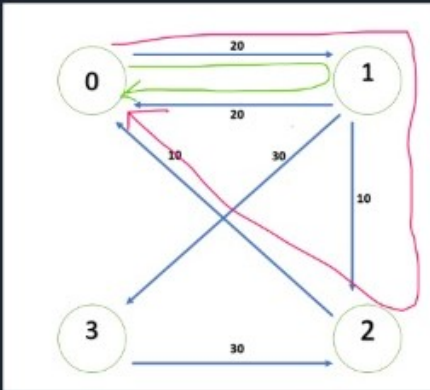
Run Code

Run Tests

Submit

33m left

## Explanation 0



For example, Let us calculate the round trip between City 1 and City 3 i.e  $R(1,3)$

$\text{Cost}(1, 3) = P(1, 3) = 30$ , because there are only one simple way to go to there.

$\text{Cost}(3, 1) = P(3, 2) + P(2, 0) + P(0, 1) = 60$ .

So the cheapest round trip from City 1 and City 3 i.e  $R(1,3)$  is equal to  $30 + 60 = 90$  units.

We can calculate all the other round trip in the similar way, then we can get the answer which is  $R(0,1)$  i.e 40.

There are 2 city pairs with the lowest cost (0,1) and (0,2).

Lowest sum of indices belongs to (0,1) & hence 0 and 1 in alphanumeric order.

## Sample Input 1

3  
3

Language

C



Autocomplete Ready



```

23  *
24  * The function is expected to return an INTEGER_ARRAY.
25  * The function accepts following parameters:
26  * 1. INTEGER NumCities
27  * 2. 2D_INTEGER_ARRAY FlightInfo
28  */
29
30 /*
31  * To return the integer array from the function, you should:
32  * - Store the size of the array to be returned in the result_count variable
33  * - Allocate the array statically or dynamically
34  *
35  * For example,
36  * int* return_integer_array_using_static_allocation(int* result_count) {
37  *     *result_count = 5;
38  *
39  *     static int a[5] = {1, 2, 3, 4, 5};
40  *
41  *     return a;
42  * }
43  *
44  * int* return_integer_array_using_dynamic_allocation(int* result_count) {
45  *     *result_count = 5;
46  *
47  *     int *a = malloc(5 * sizeof(int));
48  *
49  *     for (int i = 0; i < 5; i++) {
50  *         *(a + i) = i + 1;
51  *     }
52  *

```

Line: 20 Col:

Test Results

Custom Input

Run Code

Run Tests

Submit

33m left



ALL



35

36

37

38

39

40

41

42

For example, Let us calculate the round trip between City 1 and City 3 i.e  $R(1,3)$   
 $\text{Cost}(1, 3) = P(1, 3) = 30$ , because there is only one simple way to go to there.

$\text{Cost}(3, 1) = P(3, 2) + P(2, 0) + P(0, 1) = 60$ .

So the cheapest round trip from City 1 and City 3 i.e  $R(1,3)$  is equal to  $30 + 60 = 90$  units.

We can calculate all the other round trips in the similar way, then we can get the answer which is  $R(0,1)$  i.e 40. There are 2 city pairs with the lowest cost (0,1) and (0,2).

Lowest sum of indices belongs to (0,1) & hence 0 and 1 in alphanumeric order.

#### Sample Input 1

```
3
3
0 1 100
1 2 100
0 2 100
```

#### Sample Output 1

```
1000000
-1
-1
```

#### Explanation 1

There is no return flights from any combination of cities. Hence the answers.

Language

C



Autocomplete Ready



```
23  *
24  * The function is expected to return an INTEGER_ARRAY.
25  * The function accepts following parameters:
26  * 1. INTEGER NumCities
27  * 2. 2D_INTEGER_ARRAY FlightInfo
28  */
29
30 /*
31  * To return the integer array from the function, you should:
32  * - Store the size of the array to be returned in the result_count variable
33  * - Allocate the array statically or dynamically
34  *
35  * For example,
36  * int* return_integer_array_using_static_allocation(int* result_count) {
37  *     *result_count = 5;
38  *
39  *     static int a[5] = {1, 2, 3, 4, 5};
40  *
41  *     return a;
42  * }
43  *
44  * int* return_integer_array_using_dynamic_allocation(int* result_count) {
45  *     *result_count = 5;
46  *
47  *     int *a = malloc(5 * sizeof(int));
48  *
49  *     for (int i = 0; i < 5; i++) {
50  *         *(a + i) = i + 1;
51  *     }
52  * }
```

Line: 20 Col: 1

Test Results

Custom Input

Run Code

Run Tests

Submit