When people fly long distances, a direct city-to-city flight is often more expensive than taking two flights with a stop in a hub city. Travelers might save even more money by breaking the trip into three flights with two stops. But for the purposes of this challenge, let's assume that no one is willing to stop three times!

You have a table with individual airport-to-airport **flights**, which contains the following columns:

* id - the unique ID of the flight;
* origin - the origin city of the current flight;
* destination - the destination city of the current flight;
* cost - the cost of current flight.  
  The cities are all represented by an abbreviation composed of three uppercase English letters.

Your task is to write a select statement that lists all the cheapest possible trips that can be done in two or fewer stops. The output should have columns origin, destination, stops (indicating the number of stops in current trip), and total\_cost.

If two trips cost the same but have a different number of stops, include the one with the fewest stops. Sort the output table by origin, then by destination.

**Note**: A flight from SFO to JFK is considered to be different than a flight from JFK to SFO.

Example

For given table **flights**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **origin** | **destination** | **cost** |
| 1 | SFO | JFK | 500 |
| 2 | SFO | DFW | 200 |
| 3 | SFO | MCO | 400 |
| 4 | DFW | MCO | 100 |
| 5 | DFW | JFK | 200 |
| 6 | JFK | LHR | 1000 |

the output should be

|  |  |  |  |
| --- | --- | --- | --- |
| **origin** | **destination** | **stops** | **total\_cost** |
| DFW | JFK | 0 | 200 |
| DFW | LHR | 1 | 1200 |
| DFW | MCO | 0 | 100 |
| JFK | LHR | 0 | 1000 |
| SFO | DFW | 0 | 200 |
| SFO | JFK | 1 | 400 |
| SFO | LHR | 2 | 1400 |
| SFO | MCO | 1 | 300 |

* **[execution time limit] 10 seconds (mysql)**