Badge Progress (Details)

Points: 126 Rank: 270095

Toll Cost Digits

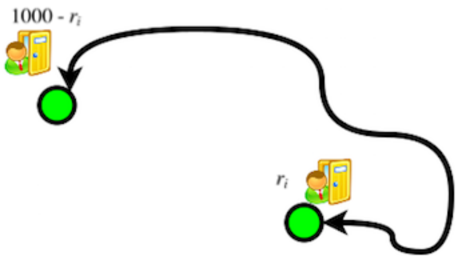
by kevinsogo

Problem | Submissions | Leaderboard | Discussions | Editorial

The mayor of Farzville is studying the city's road system to find ways of improving its traffic conditions. Farzville's road system consists of n junctions connected by e bidirectional toll roads, where the i^{th} toll road connects junctions x_i and y_i. In addition, some junctions may not be reachable from others and there may be multiple roads connecting the same pair of junctions.

Each toll road has a toll rate that's paid each time it's used. This rate varies depending on the direction of travel:

- If traveling from x_i to y_i, then the toll rate is r_i.
- If traveling from y_i to x_i, then the toll rate is $1000 - r_i$. It is guaranteed that $0 < r_i < 1000$.



For each digit $d \in \{0, 1, \ldots, 9\}$, the mayor wants to find the number of ordered pairs of (x, y) junctions such that $x \neq y$ and a path exists from x to y where the total cost of the tolls (i.e., the sum of all toll rates on the path) ends in digit d. Given a map of Farzville, can you help the mayor answer this question? For each digit d from 0 to 9, print the the number of valid ordered pairs on a new line.

Note: Each toll road can be traversed an unlimited number of times in either direction.

Input Format

The first line contains two space-separated integers describing the respective values of n (the number of junctions) and e (the number of roads).
Each line i of the e subsequent lines describes a toll road in the form of three space-separated integers, x_i, y_i, and r_i.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq e \leq 2 \cdot 10^5$
- $1 \leq x_i, y_i \leq n$
- $x_i \neq y_i$
- $0 < r_i < 1000$

Output Format

Print ten lines of output. Each line j (where $0 \leq j \leq 9$) must contain a single integer denoting the answer for $d = j$. For example, the first line must contain the answer for $d = 0$, the second line must contain the answer for $d = 1$, and so on.

Sample Input 0

```
3 3
1 3 602
1 2 256
2 3 411
```

Sample Output 0

```
0
2
1
1
2
0
2
1
1
2
```

Explanation 0

The table below depicts the distinct pairs of junctions for each d:

d	(x, y)	path	total cost
0	none		
1	$(1, 2)$	$1 \to 3 \to 2$	1191
	$(2, 3)$	$2 \to 3$	411
2	$(1, 3)$	$1 \to 3$	602
3	$(3, 1)$	$3 \to 2 \to 1$	1333
4	$(2, 1)$	$2 \to 1$	744
	$(3, 2)$	$3 \to 1 \to 2$	654
5	none		
6	$(1, 2)$	$1 \to 2$	256
	$(2, 3)$	$2 \to 1 \to 3$	1346
7	$(1, 3)$	$1 \to 2 \to 3$	667
8	$(3, 1)$	$3 \to 1$	398
9	$(2, 1)$	$2 \to 3 \to 1$	809
	$(3, 2)$	$3 \to 2$	589

Note the following:

- There may be multiple paths between each pair of junctions.
- Junctions and roads may be traversed multiple times. For example, the path $2 \to 3 \to 1 \to 2 \to 3$ is also valid, and it has total cost of $411 + 398 + 256 + 411 = 1476$.
- An ordered pair can be counted for more than one d. For example, the pair $(2, 3)$ is counted for $d = 1$ and $d = 6$.
- Each ordered pair must only be counted once for each d. For example, the paths $2 \to 1 \to 3$ and $2 \to 3 \to 1 \to 2 \to 3$ both have total costs that end in $d = 6$, but the pair $(2, 3)$ is only counted once.

Submissions:94

Max Score:60
Difficulty: Hard

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int e = in.nextInt();
        for(int a0 = 0; a0 < e; a0++){
            int x = in.nextInt();
            int y = in.nextInt();
            int r = in.nextInt();
        }
    }
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature