Algorithms and Data structures 2017: First assignment

1 Instructions

You may work in groups of two. For this assignment you have to hand in two things:

- Source code. We will run the code for grading.
- A report. In the report you have to explain the algorithm and analyse it.

Source code You are allowed to submit a solution in either C, C++, Java or Python. If you prefer another language, please contact us. You are only allowed to use the Standard Library corresponding to your selected language.
You will find a set of examples to test your solution on the course wiki. We will set up a website, on which you can upload your code. It will be automatically be tested against test cases.

Report Besides handing in code, we would like to receive a report in which you explain your algorithm and analyse its correctness and runtime complexity.

Submission The deadline for sending in your solution is on November 16. You must submit your solutions via Blackboard. Only one team member has to submit a solution; the names and student numbers of both team members must be mentioned in the report.

Grading Grades will be determined as follows. You may earn up to 100 points for your solution:

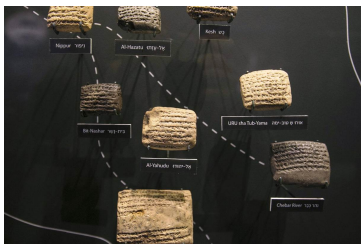
- 15 points for the explanation of your algorithm.
- 10 points for the correctness analysis.
- 10 points for the complexity analysis.
- 50 points for the test results. We will be running several tests and you will get points for every correct answer within the time limit. If your code does not compile or does not read and write via `stdin` and `stdout`, you will get zero points on the test cases. So please test your code well!
- 15 points for the quality of the code.

If you have any questions, do not hesitate to send a email to Joshua (or someone else related to the course).

2 The Constellation-City-Correspondence-Conjecture

Last year, there was big news in the field of archaeology: a new Maya city was claimed to be found by the teenager William Gadoury. [1] His claims were quickly dismissed but still show a creative way of thinking. William discovered that some of the known Maya cities actually align with some constellations (i.e. they are positioned in the same shape). Then he realised that not all stars in the constellation had a corresponding known city. Knowing that the Mayans did things for a reason, there must have been long-lost cities at these locations. In particular, he found one spot which shows a man-made structure on Google Earth, he called this city "K'aak Chi". Unfortunately for him, the man-made structure was just a field of agriculture.

Eager to discover the lost cities, William is still researching and looking for clues. He went to many museums to learn more about the Mayans and their cities.

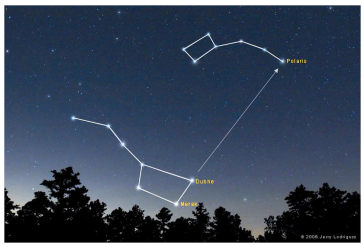


William is intrigued by these clay tableaus. Picture by Baz Ratner / Reuters.

One particular ancient piece of clay caught his attention. It is a tableau made by the Mayans containing lots of numbers. Although the museum explains that this is just a table with exchange rates, William believes that it describes something much deeper. He believes that these numbers describe the distances between the cities.

William conjectures the following: The Mayans build cities according to the stars of their constellations. Then the Mayans connect their cities with streets, also according to the constellation. In the Maya civilisation, a constellation consisted of n stars connected by n line segments. (The constellations of the Mayans are not necessarily the constellations we know in our western culture.) In the end, the Mayans construct a tableau with the shortest distances from one city to the other, using only the connections from the constellation.

William has heard that the students in Nijmegen are specialists when it comes to shortest paths, he asks you to help verify his conjecture on the constellation-city-correspondence. He wants to know which constellation possibly explains the shortest distances in the tableau.



Two well-known western constellations. Picture by Jerry Lodriguss.

Can you help William to compute which connections are needed in order to explain the tableau of shortest distances?

[1] Some more details and references can be found here: `https://www.wired.com/2016/05/long-lost-mayan-city-teen-found-isnt-lost-city/`.

2.1 Input

A tableau will be given via `stdin` (your program should not open any files) with the following format: [2]

- One line containing a single integer n (with $3 \leq n \leq 5000$), this is the number of Maya cities.
- Then n lines, each containing n integers. Let d_{ij} denote the jth integer on the ith line, then this is the shortest distance between city i to city j, and
  - $d_{ii} = 0$,
  - $1 \leq d_{ij} = d_{ji} \leq 2000000$
  - i and j are numbered from 1 to n (inclusive).

2.2 Output

You should print (to `stdout`) a constellation that gives rise to the shortest distances as listed in the tableau. More precisely:

- Print n connections as follows: for each connection between i and j of length d, print one line with three integers i, j and d, separated by spaces.

The order in which the connections are printed, does not matter. No duplicate connections are allowed. If there are multiple constellations possible, any will do. You may assume that at least one solution exists.
We will run your program and check exactly with this specification. Any other output (besides whitespace) will be regarded as a wrong output.

Example input 1:

```
4
0 1 2 3
1 0 1 2
2 1 0 1
3 2 1 0
```

Output:

```
1 2 1
3 2 1
1 3 2
3 4 1
```

Example input 2:

```
3
0 3 5
3 0 4
5 4 0
```

Output:

```
1 2 3
1 3 5
2 3 4
```

[2] You can assume the input is exactly as specified, no error-handling is required for the assignment.

Example input 3:

```
8
0   9   8   4   13  10  4   9
9   0   9   5   14  19  13  18
8   9   0   4   5   18  4   9
4   5   4   0   9   14  8   13
13  14  5   9   0   23  9   14
10  19  18  14  23  0   14  19
4   13  4   8   9   14  0   5
9   18  9   13  14  19  5   0
```

Output:

```
2 4 5
4 3 4
3 5 5
3 7 4
7 8 5
7 1 4
1 6 10
1 4 4
```