

Task Description

[Back Story](#)

[General Remark](#)

[Problem Specification](#)

[Solution Specification](#)

[The source positions part](#)

[The facets part](#)

[The destination positions part](#)

[Valid solution, Normalized solution](#)

[Contest Structure](#)

[Team Registration and Solution/Problem Submission](#)

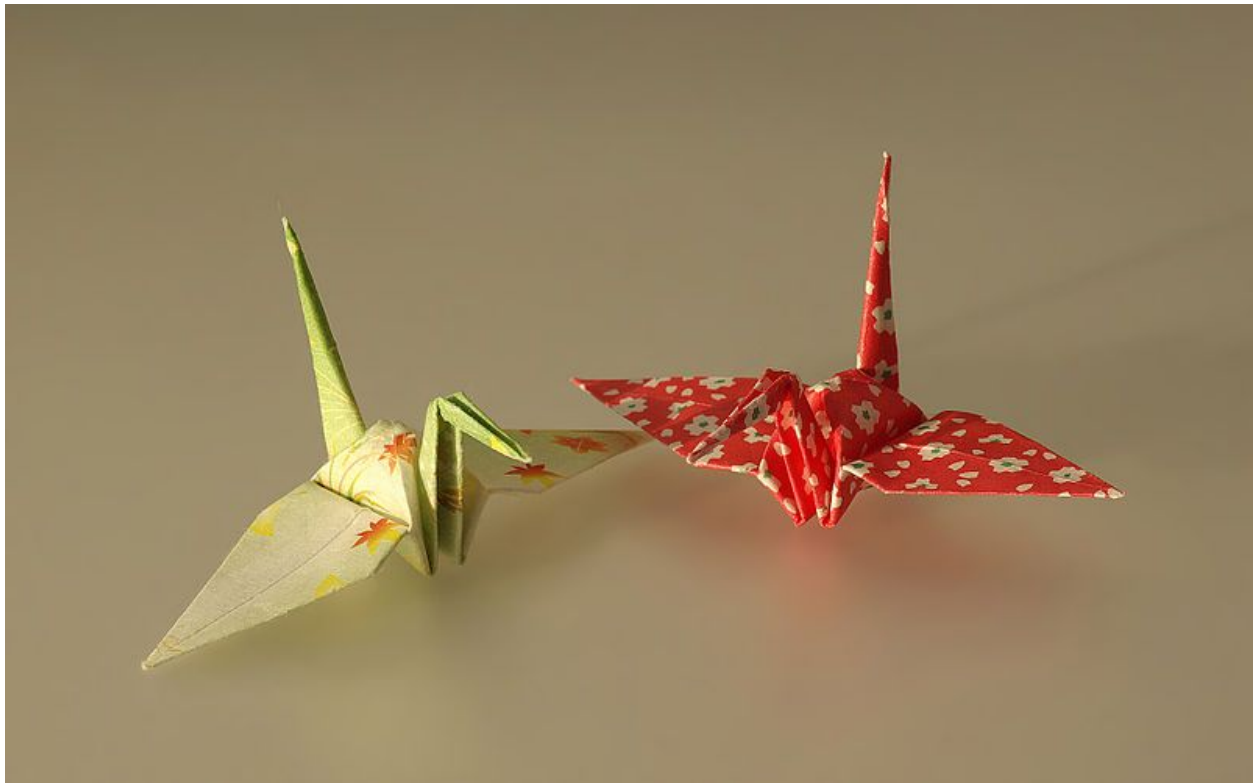
[Solution Evaluation](#)

[Scoring](#)

[Rate Limits and Leaderboard Freeze](#)

[Terms and Conditions](#)

Back Story



[Origamis](#) are artistic objects created by folding square papers into shapes, without cutting or glueing. Origami also refers to the act of creating such objects.

Origami has been practiced for a long, long time in Japan, and is now popular worldwide. But few are aware of spiritual powers of origami. They offer protection from disasters such as earthquakes, plagues and server failures. Now, we want to enhance the Big Buddha Statue of Nara, by offering enchanted origamis to the Buddha. Origami silhouettes of special meanings are described in the Kojiki, an ancient scroll, but how to fold them is lost knowledge.

As we are now in the 21st century of industry 4.0, big data, and artificial intelligence, we employ drone geishas to fold origamis. Your task is to recover the folds, each in form of a mapping from a square of paper to a special silhouette, so that the drone geishas can be programmed to fold the ancient origamis described in the Kojiki.

To make matters worse, the very act of folding origamis using drone geishas draws The Singularity near, that clouds everything (also known as IoT) and makes it impossible to see the future. In preparation for this, we need to find out more powerful origami shapes that were not even predicted in the Kojiki. We need help from all of you here; you are all hardworking

programmers, so those origami shapes that are insoluble by most of you must hold the strongest power.

We live in a two-dimensional world, so only planar origamis are considered. As long as the origami is a valid two-dimensional mapping, it is allowed to specify **impossible** folds, that require papers to penetrate each other. Never mind, one of geisha's jobs is to warp the reality into fourth dimension.

Sushi and sukiyaki will be rewarded. Wasshoi!

General Remark

In our coordinate system, the x axis points to the right, y to the up. A two-dimensional point is specified in the form $a/b, c/d$, where the numerators a, c are integers and the denominators b, d are positive integers. The $/$ and the denominator may be abbreviated when the denominator is equal to 1. Fractions may be reducible.

Often we treat polygons and edges as sets of points. In such cases the boundaries are inclusive. That is, a polygon seen as a set of points includes its edges and vertices; an edge as a set includes its two vertices.

The size of a problem file or a solution file is defined by the number of non-whitespace ASCII characters in it. This is to avoid the issue of newline-code differences among operating systems, of newline characters at the end of file, etc.

An interactive tool to help you understand the idea of the task is provided at <http://2016sv.icfpcontest.org/play>. This tool is provided just for reference; it supports only simple valley folds, so you may not be able to create some complex silhouettes with this tool. The interactive tool is not part of the task description, and the judges offer no guarantee of its consistency with the task description. The judges will not use this tool to evaluate your solutions.

Problem Specification

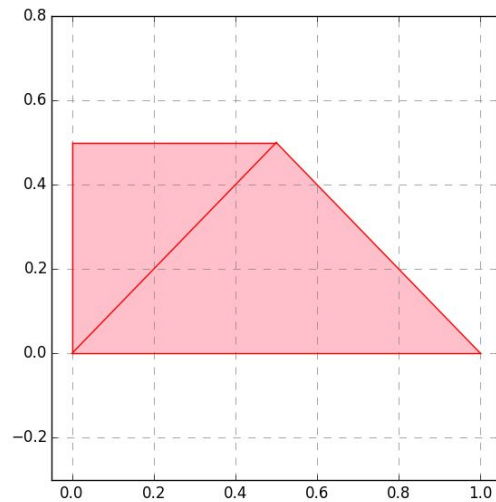


Figure 1. An origami silhouette with its skeleton.

Each problem is specified by a *silhouette* of the origami, which is a two-dimensional figure, in ASCII text format. Additionally, a *skeleton* of the origami is given as a hint. For the exact definitions of silhouette and skeleton, see the “Solution Evaluation” section.

A silhouette consists of one or more polygons. In silhouette specifications, counterclockwise polygons refer to polygons with positive area; clockwise polygons are treated as holes in the silhouette.

A silhouette specification consists of the following items, each in one line.

- the number of polygons
 - for each polygon, the number of vertices
 - for each vertex, its coordinates

A skeleton is a set of edges and folding line segments of the origami. A skeleton is specified by the following items, each in one line:

- the number of line segments
 - for each line segment, the coordinates of its two vertices, separated by a space

The skeleton specification must employ the least number of line segments; any two line segments that lie on a same line must be disjoint.

Here is a sample problem specification which corresponds to Figure 1:

```

1
4
0,0
1,0
1/2,1/2
0,1/2
5
0,0 1,0
1,0 1/2,1/2
1/2,1/2 0,1/2
0,1/2 0,0
0,0 1/2,1/2

```

For a problem to be a valid one, there must exist a **valid and normalized** solution that produces the silhouette and the skeleton exactly. The definitions for valid / normalized solutions are given in the following sections.

Solution Specification

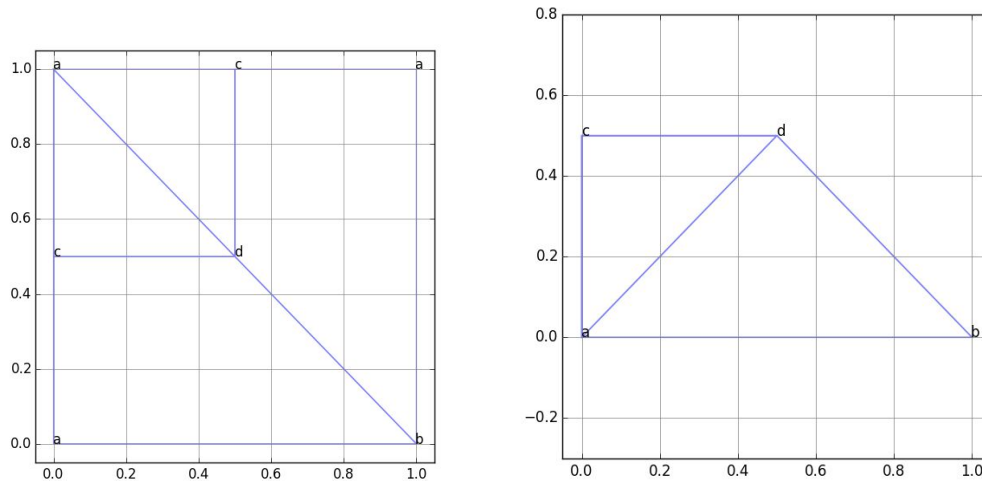


Figure 2. Visualization of facets of an origami at their source (left) and destination (right) positions. The alphabet labels represent the mapping of the vertices. This mapping constitutes a solution to the origami problem in Figure 1.

A solution to a problem is a *mapping* that produces the given silhouette. An origami consists of *facet* polygons created as results of folding the square paper. All the components of origami such as the facet polygons, their edges and vertices have two positions, one at *source* and the

other at *destination*. The source position is where the paper is a square located at (0,0), (1,0), (1,1), (0,1), and the destination position is where the paper is folded to form the silhouette.

A solution specification consists of three parts:

- the source positions part
- the facets part
- the destination positions part

The source positions part

The vertices of an origami are indexed by integers starting from 0.

The source positions part consists of the following items, each in one line:

- the number of vertices
 - for each vertex, its coordinate

The facets part

The facets part consists of the list of facet polygons. Each polygon is specified by the *indices* of the vertices. For facet polygons, the vertices can either be listed in clockwise or counterclockwise order. In both cases, facet polygons are treated as polygons with positive areas.

The facets part consists of the following items, each in one line:

- the number of facets
 - for each facet, the number of vertices, followed by the list of its vertex indices, separated by single space characters.

The destination positions part

In this part, destination coordinates of the vertices are listed, in the ascending order of the vertex index.

As an example, here is a solution that corresponds to the above example problem.

```
7
0,0
1,0
1,1
0,1
0,1/2
1/2,1/2
1/2,1
4
4 0 1 5 4
4 1 2 6 5
```

3 4 5 3
3 5 6 3
0,0
1,0
0,0
0,0
0,1/2
1/2,1/2
0,1/2

Valid solution and Normalized solution

A solution is valid if and only if it satisfies all of the following conditions:

- All the source positions of the vertices are within the initial square spanned by the four vertices $(0,0)$, $(1,0)$, $(1,1)$, $(0,1)$.
- No coordinate appears more than once in the source positions part.
- Any edge of any facet has length greater than zero.
- At source positions, if two different edges share a point, the point should always be one of the endpoints for both the edges. That is, an edge touching another edge, or edges crossing each other are prohibited.
- All facet polygons are simple; a facet polygon's perimeter must not intersect itself.
- Every facet at source position maps to its destination position, by a congruent transformation that maps its source vertices to corresponding destination vertices.
- At source position, the intersection set of any two different facets has zero area.
- At source position, the union set of all facets exactly matches the initial square.
- The size of the solution is no larger than $B_s = 5000$ Bytes.

Note that all facets are defined to have positive areas, regardless of their perimeters being clockwise/counterclockwise, as described in "The facets part" section.

A solution is normalized if and only if it satisfies all of the following conditions:

- It is valid.
- At source position, if two different facets share an edge for a length greater than 0, then the intersection set of those two facets at destination positions must have an area greater than 0. In other words, if an edge separates two facets, you should always fold the origami at that edge.

Conversely, the following properties of an origami are irrelevant to its validity and normality:

- The skeleton of the origami.
- Whether its destination silhouette fits within the source position $(0,0)$, $(1,0)$, $(1,1)$, $(0,1)$ or not.
- Whether the destination position can be reached just by folding the paper, or it requires parallel transformation and/or rotation of the paper to be reached.
- Whether the paper is not folded at all.

Contest Structure

Team Registration and Solution/Problem Submission

In this section we explain the structure and schedule of the contest. We use the Contest Clock to refer to the timings of various events in the contest. The Contest Clock is set to 00:00 (CC) at the start of the contest, which is 00:00 (UTC) 5 August, 2016. The contest ends at 72:00 (CC).

At 00:00 (CC) you can start team registration on <http://2016sv.icfpcontest.org/>. You need to register your team to view problems and submit solutions.

The contest consists of two rounds. The first 24 hours (00:00 (CC) - 24:00 (CC)) is the Lightning Round where contestants can only submit solutions to those problems that are described in the Kojiki. You don't need to submit your source code at the end of the Lightning Round.

The following 48 hours (24:00 (CC) - 72:00 (CC)) is the Full Round where contestants can also view and solve problems created by other contestants. At the end of the Full Round, you will be asked to submit your source code.

Every one hour, the contest server reveals one problem per each team, starting at 24:00 (CC) and till 69:00 (CC). Therefore, each team can publish a maximum of 46 problems. You can pre-register the problems you want to publish, and the contest server automatically publishes them in the order you specified. Pre-registration of the problems starts at 00:00 (CC).

Remember, you must submit a **valid and normalized solution** that produces the problem for your problem submission to be accepted. The problem specification will be automatically generated from the solution you have submitted.

A team cannot submit a solution to a problem submitted by itself.

Solution Evaluation

For an origami solution, its *silhouette* is the union set of all its facets at the destination position. The *skeleton* of a solution is the union set of all the facet edges at the destination positions. Our goal is to approximate, and if possible, reproduce the problem silhouette using your solutions. The skeleton in the problem specification is just a hint, therefore the skeleton of a solution is irrelevant for scoring.

Solutions submitted for a problem are ranked by *resemblance* of their silhouette to the problem silhouette. The resemblance of a valid solution is calculated by the following formula:

$$\text{resemblance} = \text{area_and} / \text{area_or}$$

- `area_and` is the area of the intersection set of the solution silhouette and the problem silhouette.
- `area_or` is the area of the union set of the solution silhouette and the problem silhouette.

Resemblance of each solution is rounded down to six decimal places. Therefore, the highest resemblance attainable by an approximate solution is 0.999999 .

Scoring

Teams are ranked by *team score*. Teams can increase their scores both by submitting problems and solutions. The team score is the sum of the *problem score* and the *solution score* the team receives, calculated as following.

Submitting a *perfect solution*, that is, a solution with resemblance = 1.0, is important in making high scores. For a problem, let s be the size of the problem, and n be the number of the teams that submitted a perfect solution, plus 1. The problem-setting team receives $(5000 - s) / n$ points, and each team that submitted a perfect solution gets s / n points. The scores for the teams with imperfect solutions are calculated such that the total points earned by all the imperfect solution teams is s / n , and each team's share is proportional to the resemblance of its solution.

Remember, you cannot submit a solution to your own problems.

For example: when 39 teams submitted a perfect solution to a 1000-bytes problem, the problem-setting team will earn $(5000 - 1000) / (39 + 1) = 100.0$ points. Each team that submitted a perfect solution will earn $1000 / (39 + 1) = 25.0$ points, and each team that submitted an imperfect solution will share the total of $1000 / (39 + 1) = 25.0$ points.

If you submit multiple solutions to one problem, only the best one is used for scoring. That is, the solution with the highest resemblance, and in case of a tie, the solution with the smallest size.

The team with the highest team score is the winner of the contest. In case of a tie, the sum of the sizes of the scoring solutions is calculated for each team. The team with smaller total size will win.

Rate Limits and Leaderboard Freeze

A team may not make more than 1000 submissions per hour (both the problem and the solution submissions count.) That is, a team can make up to 1000 submissions during 00:00 (CC) - 01:00 (CC), 1000 submissions during 01:00 (CC) - 02:00 (CC), and so on. Also, the period between two consecutive submissions must be greater than 1 second. The judges will update the problem list and the ranking every hour, so kindly refrain from refreshing too frequently. The

leaderboard will be updated till 66:00 (CC); after that the leaderboard will be frozen, and the final results will be published at ICFP 2016. The statistics for each problem will be updated till the end of the contest.

Clarifications and Questions

Please submit your clarifications and questions as comments to this blog article. In order to provide consistent answers to all contestants, and to make the questions and answers easier to find, we kindly wish your cooperation in using this method of asking questions.

Terms and Conditions

By registration, ICFP programming contest (ICFPC) participants understand and agree to the following terms and conditions.

Contest participants retain ownership of all intellectual property rights in and to any submitted solutions, source codes, custom tools, and related materials ("submissions") that participants had before submission. As a condition of entry, participants grant ICFPC judges a non-exclusive, perpetual, irrevocable, worldwide, royalty-free license to use, reproduce, publish, distribute, publicly perform, and publicly display the submissions for the purposes of allowing ICFPC judges to test and evaluate the submissions for purposes of the contest.

One person may only be member of a single team, and teams may not divide or collaborate with each other once the contest has begun. As long as contest participants follow these terms and conditions, and applicable laws, there is no limitation to the number of members in a single team, and contest participants may use whatever programming languages and computer resources.

Although the judges take the best effort to keep the contest server running, the server is not guaranteed to be available all time throughout the contest. In particular, the server may be temporarily unavailable, or available at lowered submission rate limits, due to server maintenance, server resource constraints or other reasons. Judges are not obliged to take any compensation measures in such cases.

The judges retain the right to monitor, record, and investigate the submissions, other contest-related activities, or lack thereof, of participants. The records are used for the sole purpose of judgement and are discarded once the contest-related events are over. Violations of these terms and conditions, any attempts to compromise the integrity of the contest infrastructure, attempts to interfere with other participants, attempts to spoil the joy and spirit of

the contest will lead to disqualification of the involved participants and revocation of the related scores and prizes. The decisions of the judges on these matters are final. There is no right of appeal. In case of dispute, the judges retain the right to send ninjas to you to “resolve” the issue.