6.849: Geometric Folding Algorithms

Fall 2012 — Prof. Erik Demaine

Problem Set 1 Solutions

We will drop (ignore) your lowest score on any one problem.

Problem 1. Pose any problem related to folding (mathematical, computational, application, etc.). The problem should be original to the best of your knowledge (though it's fine if it turns out that the problem has been posed or even solved).

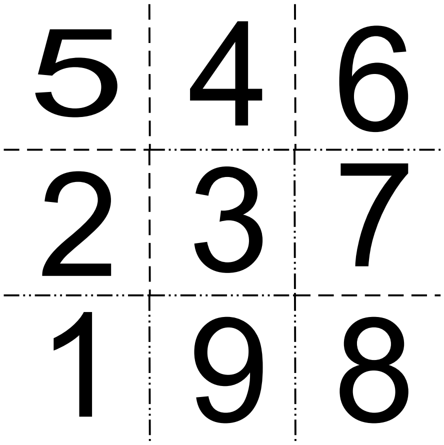
Solution: There is no one correct answer.

Problem 2. For each of the two 3×3 maps linked below,[1] fold along all the creases to obtain a flat packet of 1×1 squares that satisfy the following restrictions. Submit your folded solutions.

(a) Respect the specified mountain-valley pattern.

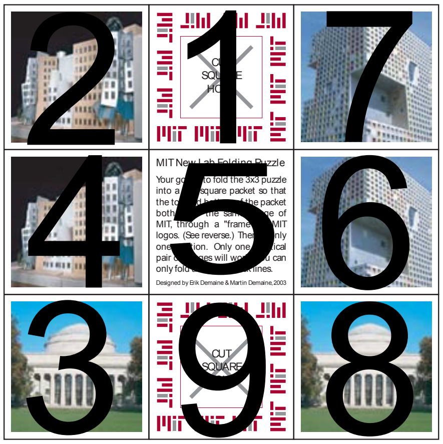
(b) The top and bottom of the packet should both show the same image of an MIT building, through a "frame" of MIT logos. (Only one identical pair of images will work.)

Solution: Layering given below. Stata appears in the windows.



(a)

http://courses.csail.mit.edu/6.849/fall12/psets/map_crease_2.pdf


2
1
7
4
5
6
3
9
8
MIT New Lab Folding Puzzle
CUT SQUARE
Designed by Erik Demaine & Martin Demaine, 2003

(b)

http://erikdemaine.org/puzzles/LCS2003/

[1] The puzzle in 2(b) was handed out in class. If you missed it, follow the link and print the two-sided PDF on a duplex printer. Note that the puzzle has two sides; only the front side is shown here.

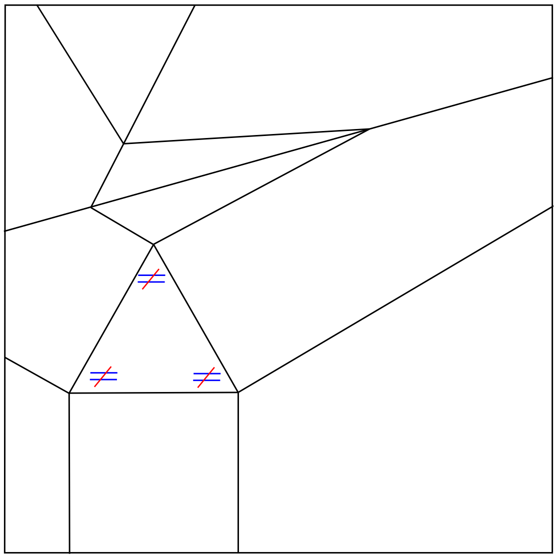
Problem 3. Give a polynomial-time algorithm to compute the minimum number of simple folds needed to fold a given 2D map (mountain-valley pattern), assuming that the map is already known to be flat foldable by simple folds.

Solution: We intended this problem to ask about all-layers simple-folds. We apologize for writing it ambiguously. For one-layer simple folds, we must make exactly one fold for every crease, and so minimizing the number of folds is trivial. For some-layers simple folds, the problem appears to be open. This problem seems to be difficult because the layering of the folds affects what folds are legal in the future and we don't always need to ensure creases line up. This seems to lead to an exponential number of cases that need to be checked. Special thanks to Brian Hamrick who pointed out the ambiguity and gave clear arguments on both sides.

For the case of all-layers simple folds, we can solve it in the following manner. A crease line is a crease that extends all the way across the rectangle and is either all mountain or all valley. We know that to be flat foldable, at least one crease line must exist, and by Lemma 14.2.1 in GFALOP we know that all crease lines in a given orientation must be folded before any other creases. Thus we can reduce the problem to a series of one-dimensional subproblems which we will call *sections*.

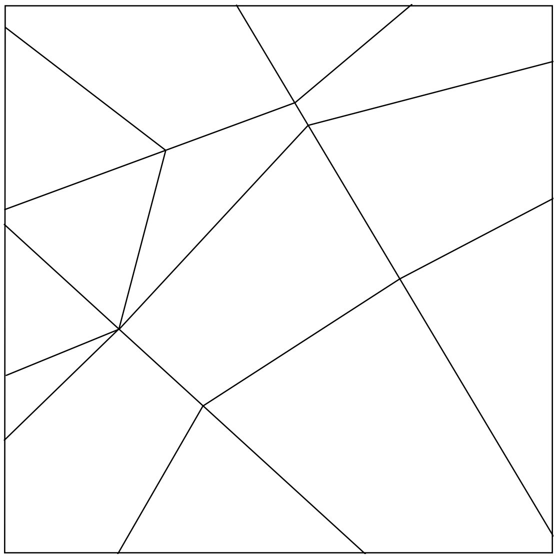
Now, for the one-dimensional case, we notice that once a fold is make, it can never be unfolded in this model. Thus, for a fold to be legal, the crease pattern must match on both the top and bottom. Further, after a fold is made, we can imagine the paper 'fusing' together, because it must all be folded at the same time. This has two important implications. First, the order in which creases are folded in each section does not change what folds are possible in future sections. Because this is the case, it suffices to solve each section optimally. Second, every time we make a fold, the resulting one-dimensional subproblem has at least one less crease than the original. This leaves $O(n^2)$ subproblems. Each of these subproblems can have at most $O(n)$ creases to fold. Thus, we can then use dynamic programming to check how many creases are needed and to choose the sequence that minimizes this number with only $O(n^4)$ operations. This running time can be reduced further with more careful handling of the dynamic programming and crease condition checking, but already achieves our polynomial-time goal.

Problem 4. Which of the following crease patterns are flat foldable? Are any simply foldable (foldable by a sequence of simple folds)? Justify each answer by either submitting a flat folding or arguing why the crease pattern cannot fold flat.



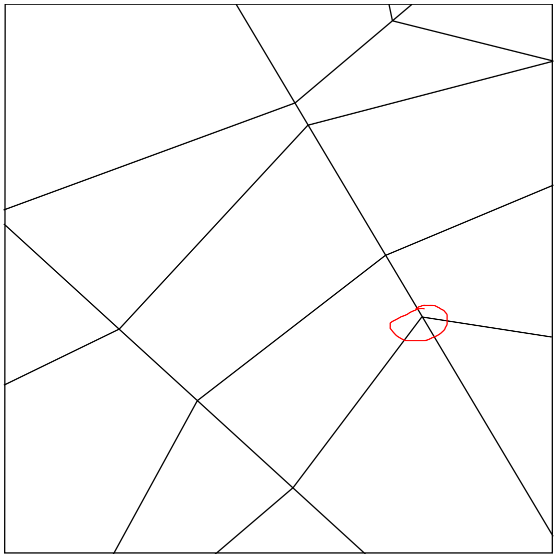
(a)

http://courses.csail.mit.edu/6.849/fall12/psets/ps1a.pdf



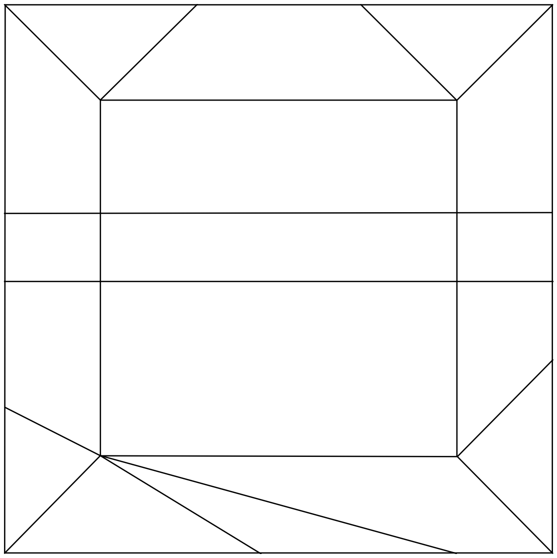
(b)

http://courses.csail.mit.edu/6.849/fall12/psets/ps1b.pdf



(c)

http://courses.csail.mit.edu/6.849/fall12/psets/ps1c.pdf



(d)

http://courses.csail.mit.edu/6.849/fall12/psets/ps1d.pdf

(a) The middle triangle violates Bern and Hayes. All pairs of edges must have opposite MV assignments which is impossible.

(b) This was intended to be flat foldable by simple folds. Violation of Kawasaki's theorm is also true and acceptable.

(c) The lower-right vertex violates Kawasaki's theorem.

(d) This is flat foldable, but not by simple folds.