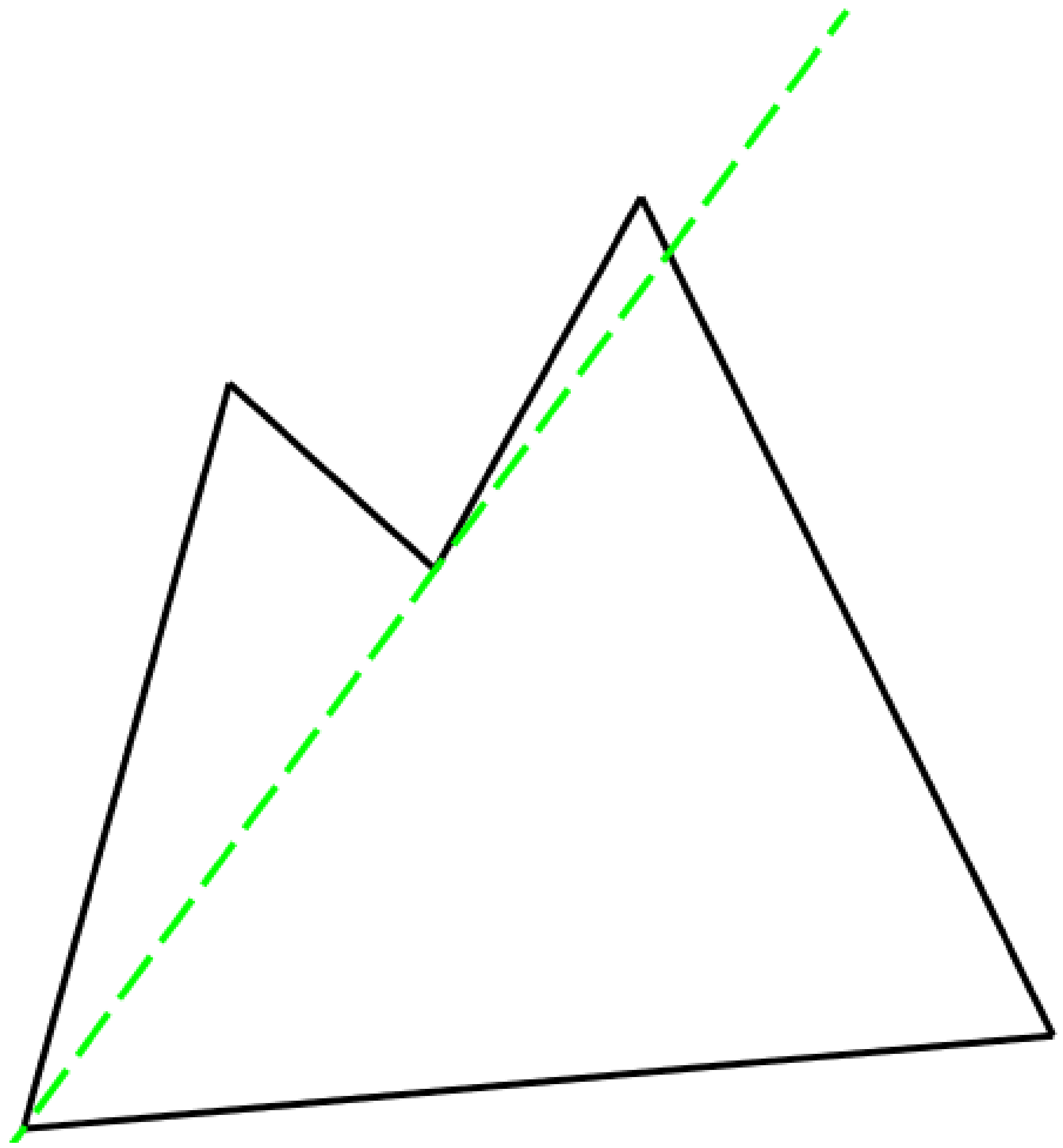
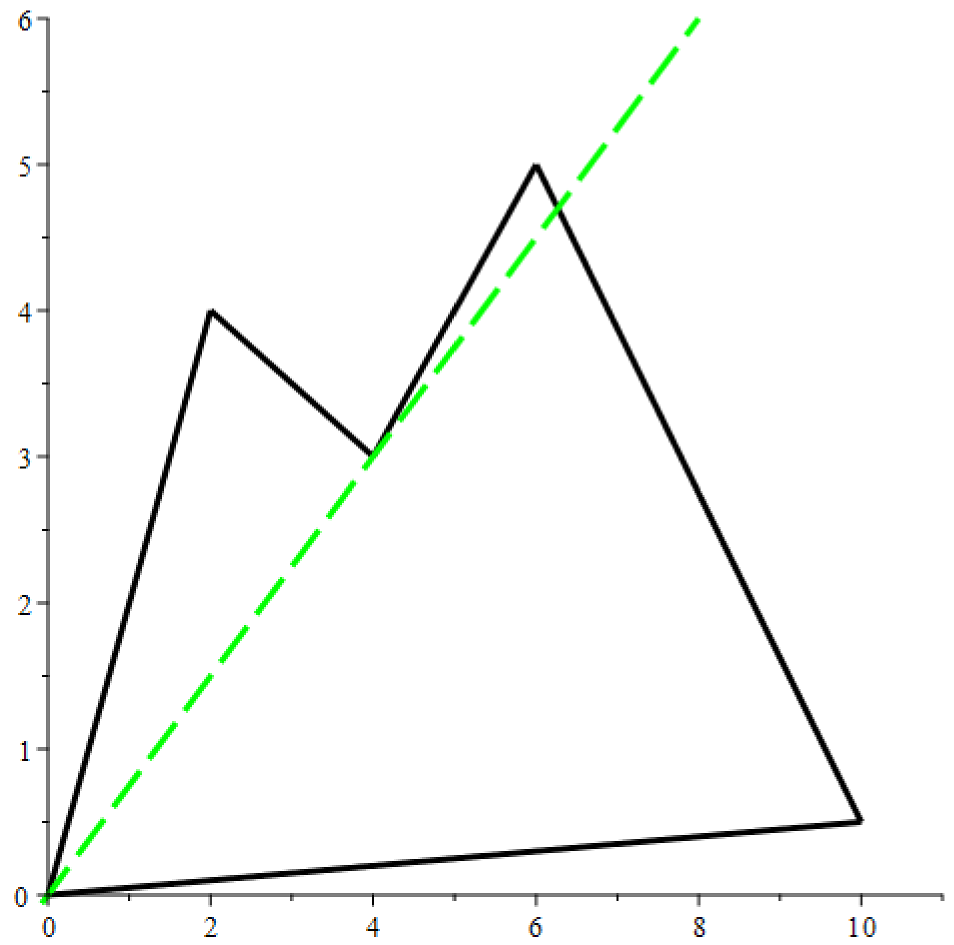
Polygons and a hash function

Robin Whitty
LSBU Maths Study Group
14th September 2023



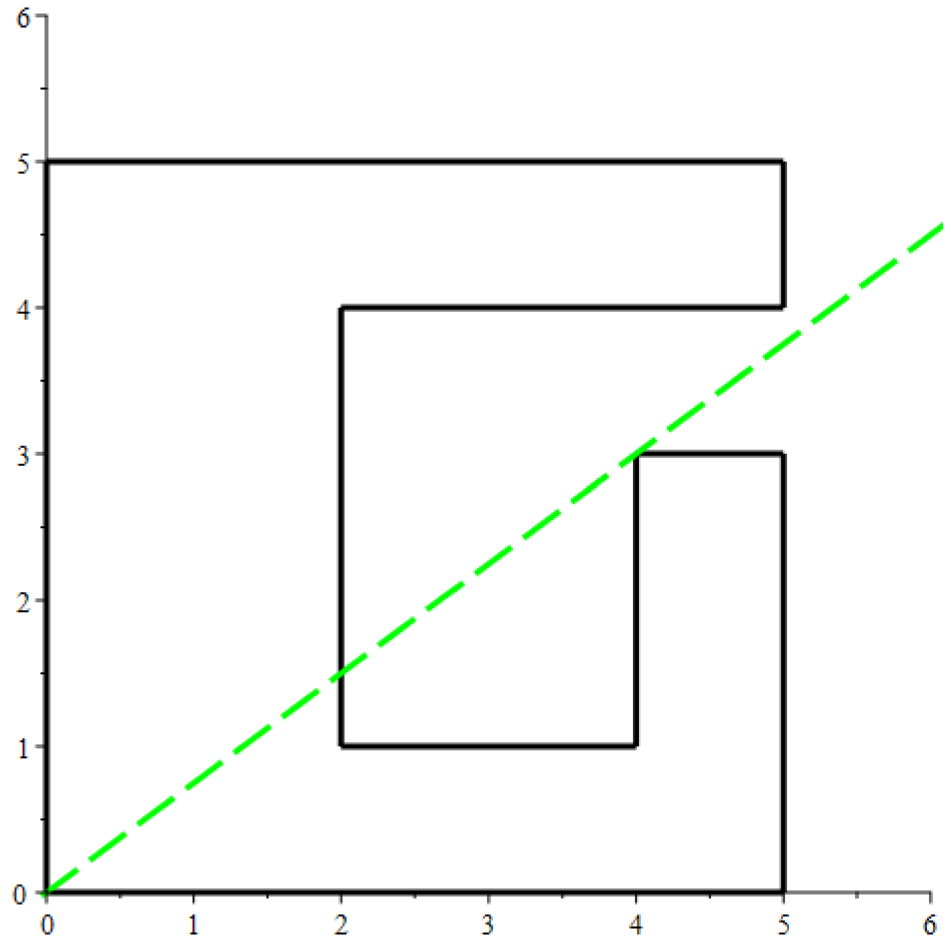
That old Intermediate Value Theorem game

Aim: translate green vector to give a chord bisecting the area of the polygon

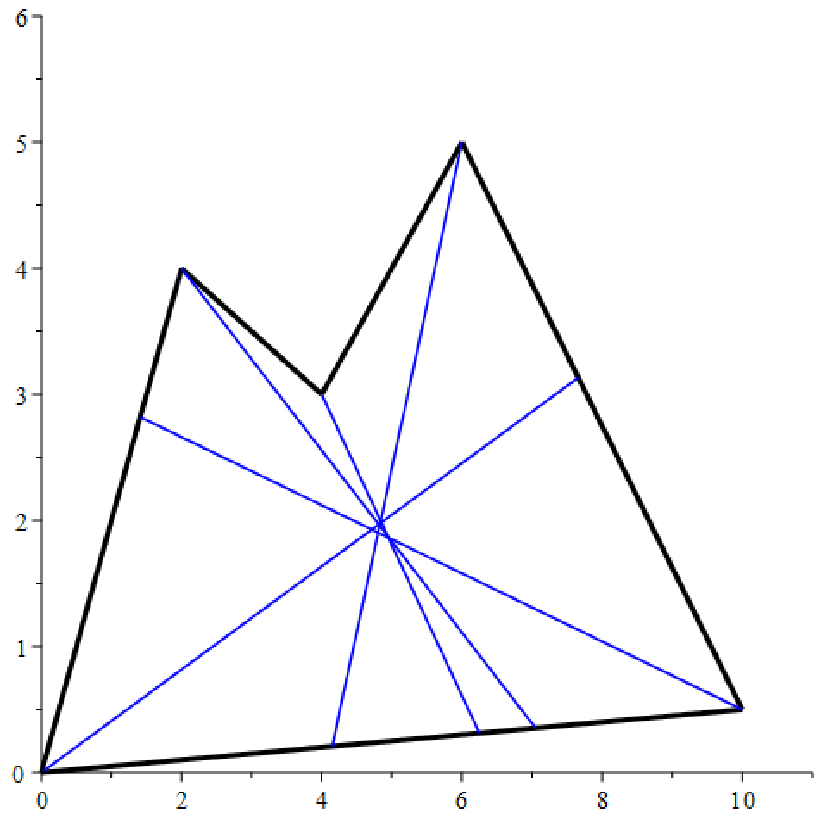


Intermediate Value Theorem applies but…

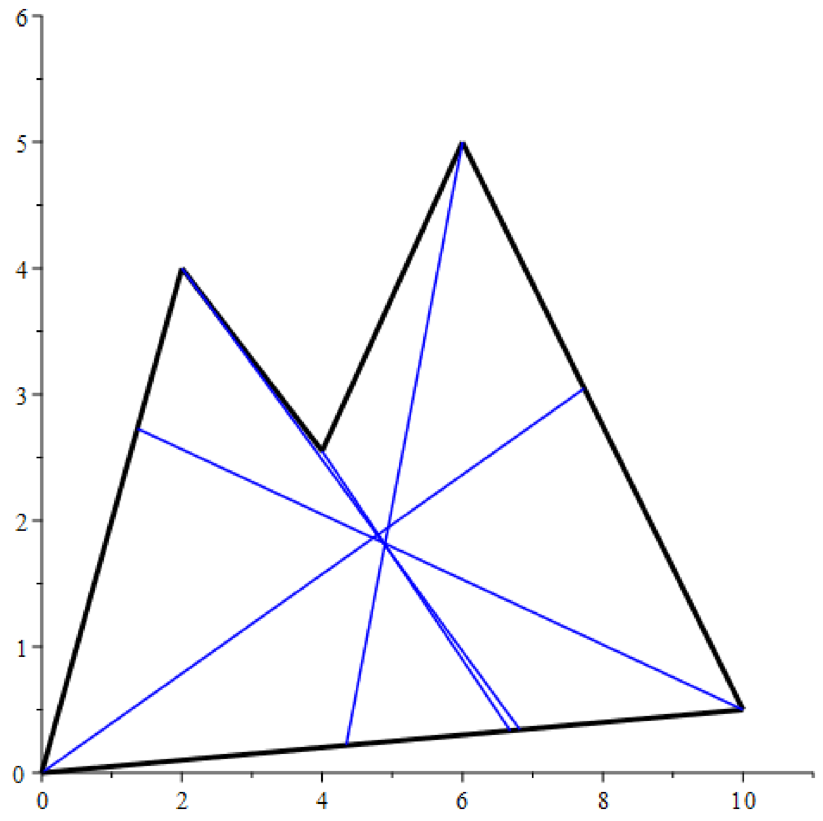
Way beyond me to systematically translate vector to find the right ‘chord’



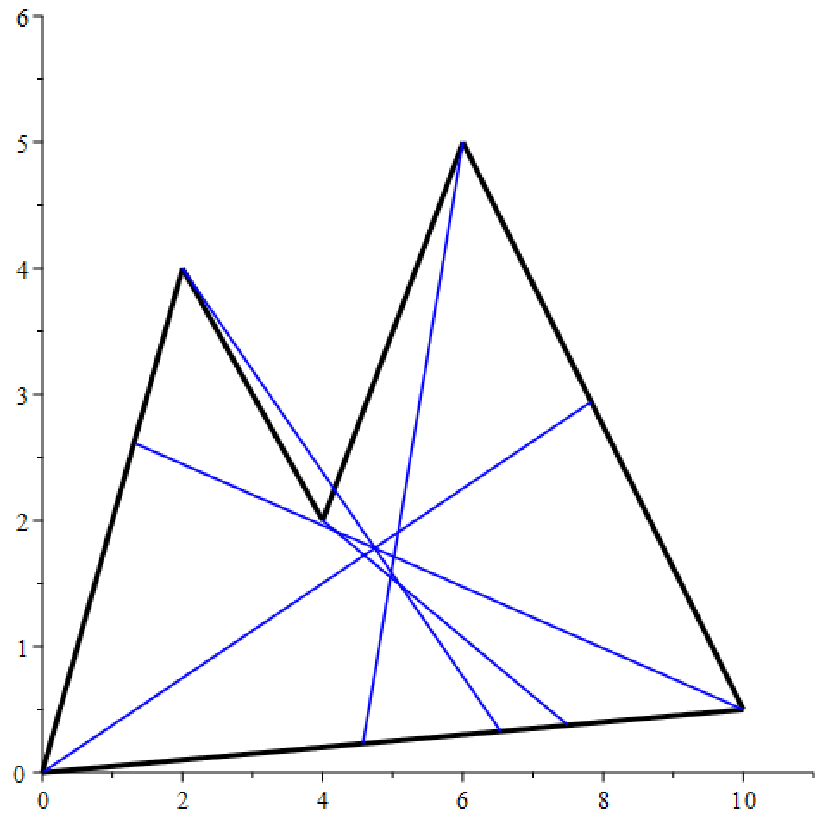
Our terms of engagement:



Strictly bisection convex: all bisecting straight lines are distinct chords



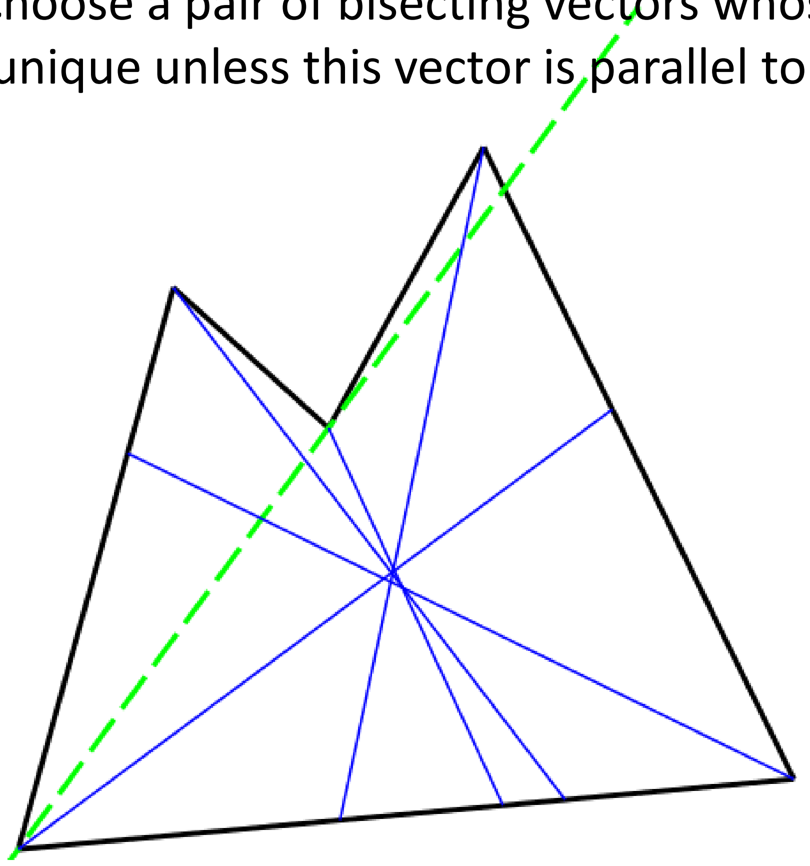
Non-strictly bisection convex: all bisecting straight lines are not-necessarily distinct chords

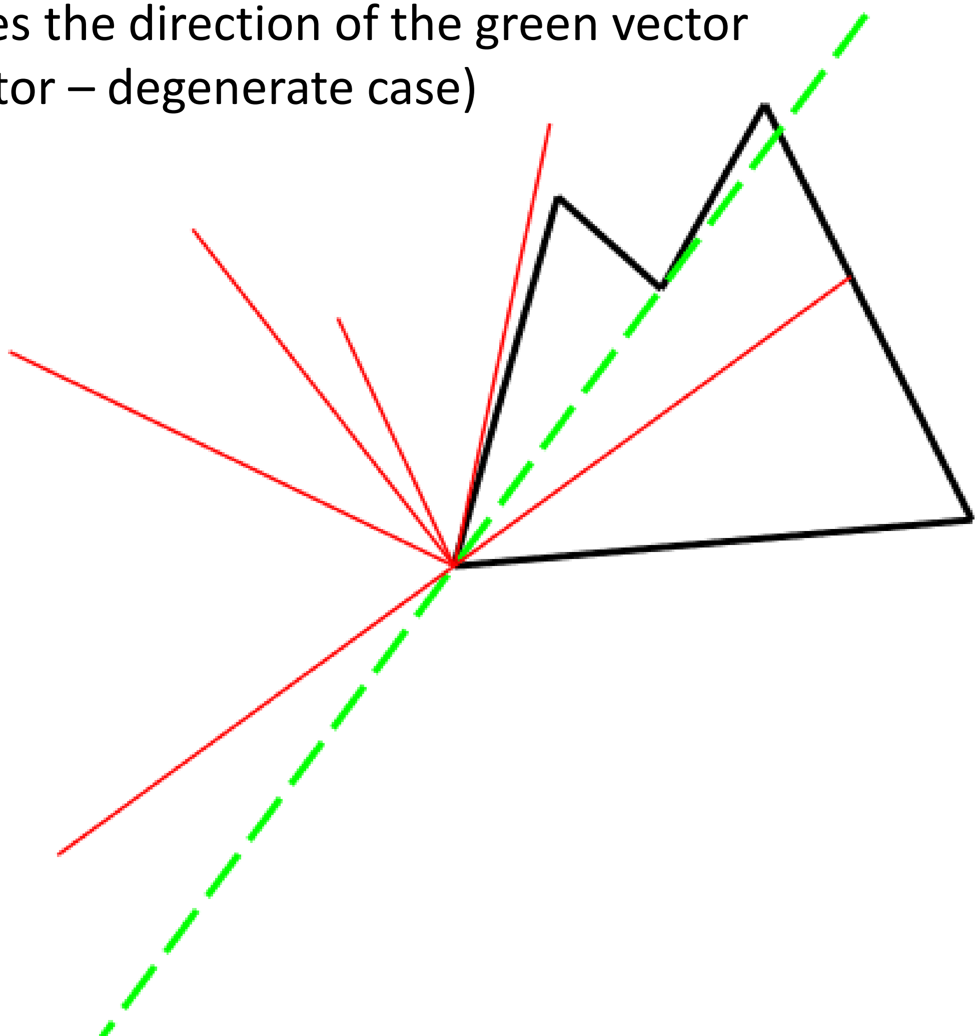


Not bisection convex: some bisecting straight lines cross polygon edges

Our approach (for bisection convex polygons):

- Find a bisecting line from each vertex to an opposite edge.
- Form a corresponding collection of anti-clockwise consecutive direction vectors which cover a half-circle
- Choose a pair of bisecting vectors whose angle includes the direction of the green vector (unique unless this vector is parallel to a bisecting vector – degenerate case)

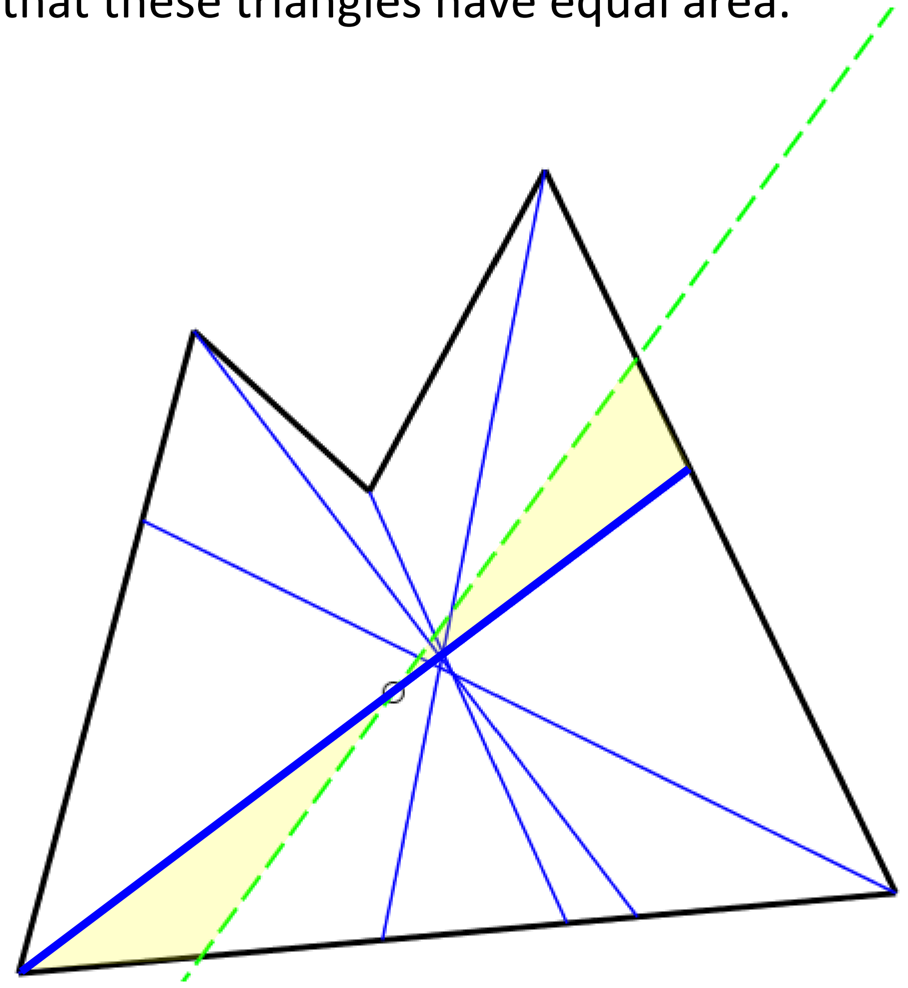


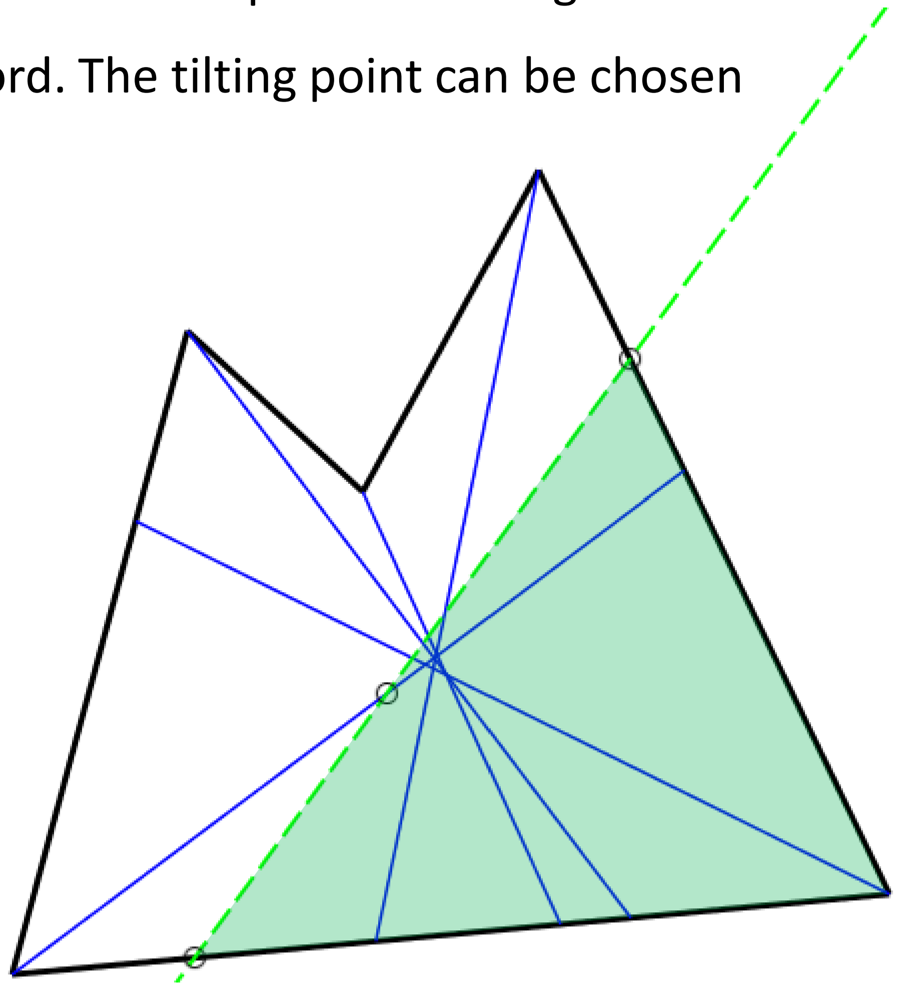


The tilting trick

Tilt anti-clockwise from the first bisecting chord to give the angle of our required bisecting vector.

By construction this will form two triangles on the bisecting chord. The tilting point can be chosen so that these triangles have equal area.





We can confirm that the 'half-polygon' formed by the tilted chord has half the original area.

Residual issues

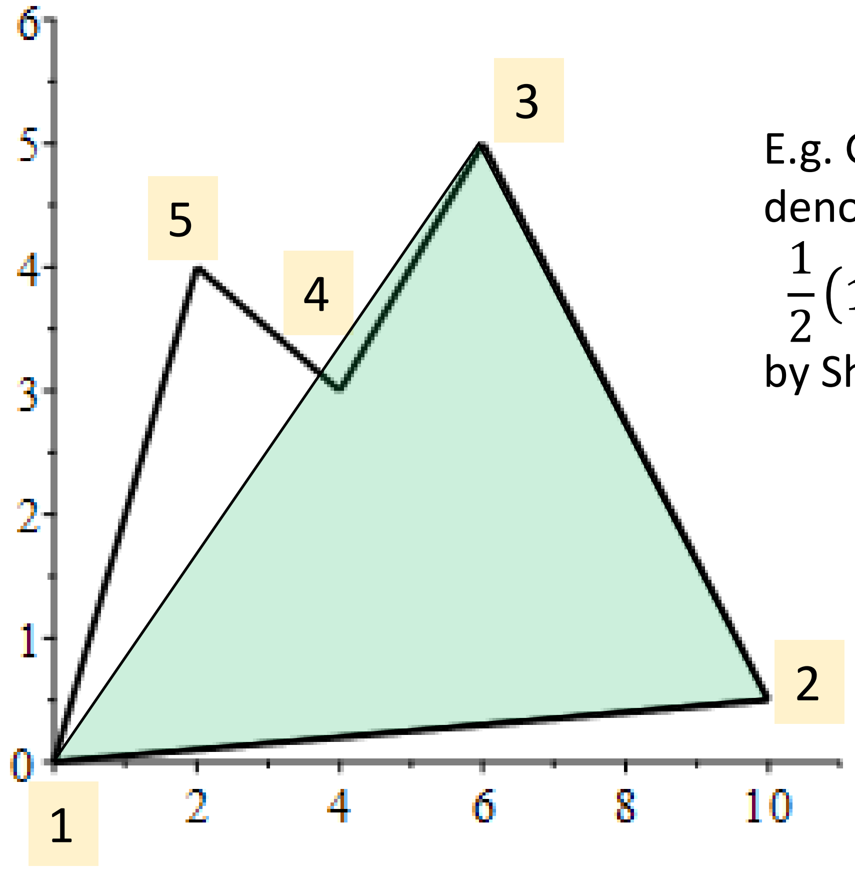
- Finding the bisecting chords between which to locate the required vector involves forming a matrix of triangle areas. This is a puzzling (to me) property.
- The triangle areas can be used to construct the bisecting chords systematically using a feed-forward mechanism which seems suggestive (to me).
- Ordering the bisecting chords to make them consecutive around a half-circle can be done by taking walks on a torus. This seems of independent interest (to me).

The characteristic polynomial puzzle

Characteristic polynomial of triangle areas matrix for n-vertex polygon with area P is (apparently)

$$\det(A - qI) = q^{n-3}(q - P)(q + P/2 \pm \alpha i)$$

What is α?



E.g. Green triangle area denoted Δ_{12} is $\frac{1}{2}\left(10 \times 5 - 6 \times \frac{1}{2}\right) = {}^{47}/_{2}$ by Shoelace formula.

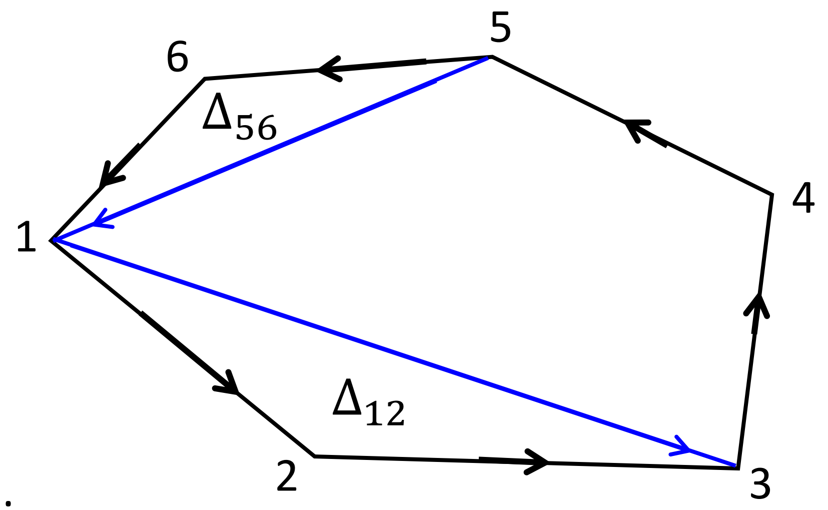
Triangle areas matrix $(\Delta_{ij})_{\substack{i=1\ldots n \\ j=i+1\ldots n-2+i}}$

$$\begin{bmatrix} 0 & \frac{47}{2} & -1 & 5 & 0 \\ 0 & 0 & \frac{17}{2} & -\frac{1}{2} & \frac{39}{2} \\ \frac{47}{2} & 0 & 0 & -3 & 7 \\ 14 & \frac{17}{2} & 0 & 0 & 5 \\ \frac{39}{2} & 11 & -3 & 0 & 0 \end{bmatrix}$$

Char poly: $q\left(q - \frac{55}{2}\right)\left(q + \frac{55}{4} \pm \frac{\sqrt{5303}}{4} i\right)$

Bisecting chords from the matrix of triangle areas

t_{ij} is the proportion in which edge $j, j+1$ must be divided in order for the chord from vertex i to be bisecting. This proportion must lie in $[0,1]$ for it to actual be a chord.



	1	2	3	4	5	6
1		Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	
2			Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}
3	Δ_{31}			Δ_{34}	Δ_{35}	Δ_{36}
4	Δ_{41}	Δ_{42}			Δ_{45}	Δ_{46}
5	Δ_{51}	Δ_{52}	Δ_{53}			Δ_{56}
6	Δ_{61}	Δ_{62}	Δ_{63}	Δ_{64}		

$$t_{15} = 1 - \frac{P}{2\Delta_{15}} \qquad t_{26} = 1 - t_{12}\frac{\Delta_{12}}{\Delta_{26}} \qquad t_{31} = 1 - t_{23}\frac{\Delta_{23}}{\Delta_{31}} \qquad t_{42} = 1 - t_{34}\frac{\Delta_{34}}{\Delta_{42}}$$

$$t_{14} = 1 + t_{15}\frac{\Delta_{15}}{\Delta_{14}} \qquad t_{25} = 1 + t_{26}\frac{\Delta_{26}}{\Delta_{25}} \qquad t_{36} = 1 + t_{31}\frac{\Delta_{31}}{\Delta_{36}} \qquad \vdots$$

$$\cdots$$

$$t_{13} = 1 + t_{14}\frac{\Delta_{14}}{\Delta_{13}} \qquad t_{24} = 1 + t_{25}\frac{\Delta_{25}}{\Delta_{24}} \qquad t_{35} = 1 + t_{36}\frac{\Delta_{36}}{\Delta_{35}}$$

$$t_{12} = 1 + t_{13}\frac{\Delta_{13}}{\Delta_{12}} \qquad t_{23} = 1 + t_{24}\frac{\Delta_{24}}{\Delta_{23}} \qquad t_{34} = 1 + t_{35}\frac{\Delta_{35}}{\Delta_{34}}$$

A feed-forward computation

We define a series of values f_n in terms of a constant K and a list of lists of values $\left(l_{ij}\right)$:

$$f_{n+1} = K + f_n \times l_{ij},$$

or $f_{n+1} = K - f_n \times l_{ij}$, if l_{ij}

if l_{ij} is the first entry in one of the lists.

In our application the f_n are the t_{ij}, $K = 1$ and the $\left(l_{ij}\right)$ are ratios of triangle areas

$$t_{15} = 1 - \frac{P}{2\Delta_{15}} \qquad t_{26} = 1 - t_{12}\frac{\Delta_{12}}{\Delta_{26}}$$

$$t_{14} = 1 + t_{15}\frac{\Delta_{15}}{\Delta_{14}}$$

$$t_{13} = 1 + t_{14}\frac{\Delta_{14}}{\Delta_{13}}$$

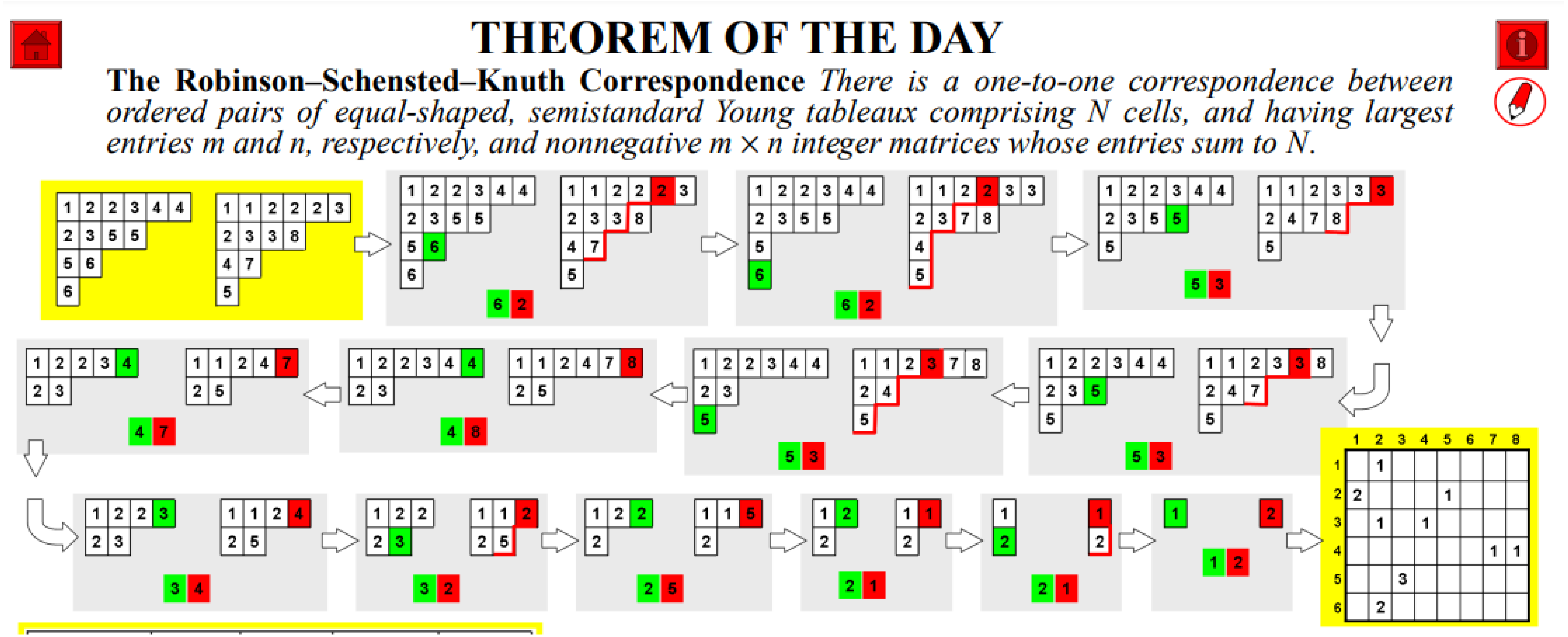
$$t_{12} = 1 + t_{13}\frac{\Delta_{13}}{\Delta_{12}}$$

(The $\left(l_{ij}\right)$ don't really form a matrix because some of the triangle areas Δ_{ij} may be zero and are omitted from the calculation of the t_{ij})

	1	2	3	4	5	6
1		Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	
2			Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}
3	Δ_{31}			Δ_{34}	Δ_{35}	Δ_{36}
4	Δ_{41}	Δ_{42}			Δ_{45}	Δ_{46}
5	Δ_{51}	Δ_{52}	Δ_{53}			Δ_{56}
6	Δ_{61}	Δ_{62}	Δ_{63}	Δ_{64}		

Feed-forward looking for an application…

Where else might this feed-forward apply naturally? E.g. Young tableaux are lists of lists of numbers.



THEOREM OF THE DAY

The Robinson–Schensted–Knuth Correspondence *There is a one-to-one correspondence between ordered pairs of equal-shaped, semistandard Young tableaux comprising N cells, and having largest entries m and n, respectively, and nonnegative m × n integer matrices whose entries sum to N.*

… in cryptography?

Feed-forward mechanisms give a way of scrambling input values, or producing pseudo-random output from an input seed value.

An example might be to calculate a hash value for a password.

A **hash function** f maps, say, a password p, non-injectively, to a hash value $h = f(p)$, so that it is hard (for an attacker)

1. To find p for a given h value (f is a one-way or 'trap door' function)
2. Given p to find p' such that $f(p') = f(p)$, (creating 'collisions')

This should be true even if everything about the function f is known to the attacker.

So how does $f(x) = K + x \times l_{ij}$ do as a hash function?

Terribly!

My hash function exploded

Hash function $f(x) = K + x \times l_{ij}$.

E.g. $(l_{ij}) = (\mathrm{a}, \mathrm{b}, \mathrm{c}), (\mathrm{d}, \mathrm{e}, \mathrm{f}, \mathrm{g}), \dots$

$$x \longrightarrow K + ax \longrightarrow K + (K + xa)b \longrightarrow K + (K + (K + xa)b)c \longrightarrow \cdots$$

Run $f(x) = K + x \times l_{ij}$ on the output but using lists …, $\left(\frac{1}{c}, \frac{1}{b}, \frac{1}{a}\right)$.

$$K + (K + (K + xa)b)c \longrightarrow K + \frac{K}{c} + K + (K + xa)b \longrightarrow K + \frac{2K}{b} + \frac{K}{bc} + K + xa$$
$$\longrightarrow \frac{2K}{a} + \frac{2K}{ab} + \frac{K}{abc} + x$$

Since we know everything about f we can remove the values of K, a, b, c and we are left with the original input x.

But it’s worse!

My hash function explodes interestingly (to me)

I took the hash function and looked at all possible sign changes for the reciprocals in the attack just described:

$$hh := (((((((s\,a\_ + q)\,b\_ + q)\,c\_ + q)\,d\_ + q)\,e\_ + q)\,f\_ + q)\,g\_ + q)\,h\_ + q$$

(I used s and q instead of x and K – I was too lazy to re-run the calculation to make it consistent)

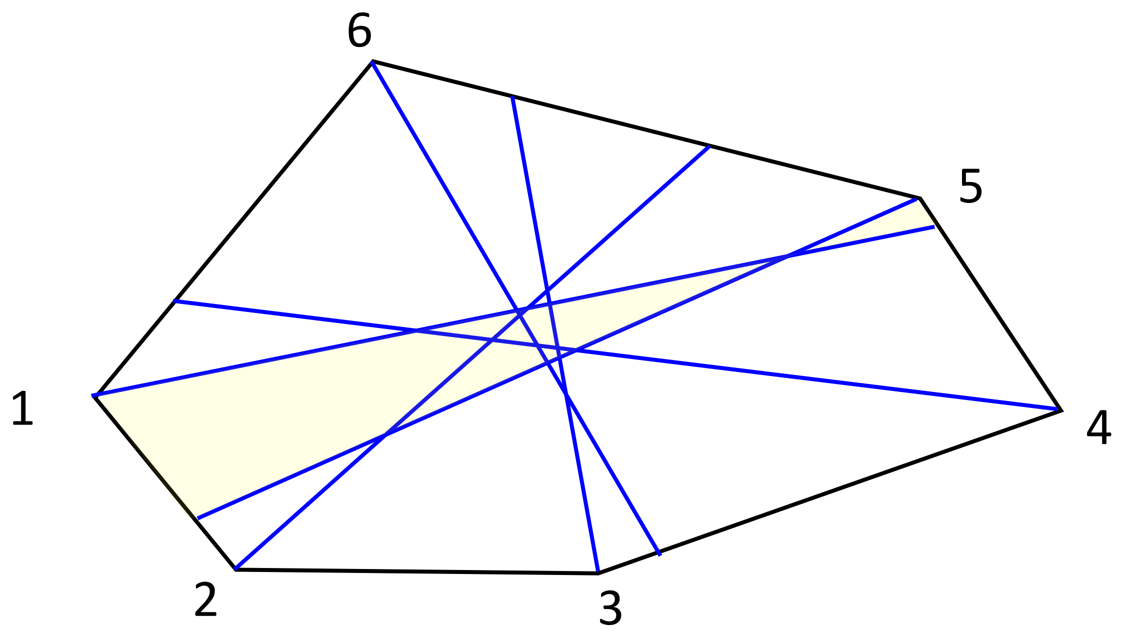
Regardless of the lengths of lists, the same two choices of signs uniquely cause everything to cancel except for two additional terms:

Output of reversed function:

$$\left[-\frac{1}{h\_}, \frac{1}{g\_}, \frac{1}{f\_}, \frac{1}{e\_}, \frac{1}{d\_}, \frac{1}{c\_}, \frac{1}{b\_}, \frac{1}{a\_}\right], q - s - \frac{q}{h\_\,g\_\,f\_\,e\_\,d\_\,c\_\,b\_\,a\_}$$

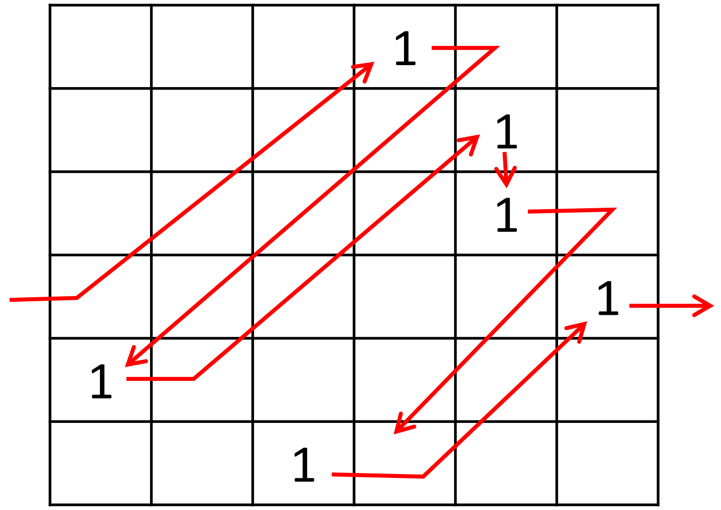
$$\left[-\frac{1}{h\_}, \frac{1}{g\_}, \frac{1}{f\_}, \frac{1}{e\_}, \frac{1}{d\_}, \frac{1}{c\_}, \frac{1}{b\_}, -\frac{1}{a\_}\right], q + s + \frac{q}{h\_\,g\_\,f\_\,e\_\,d\_\,c\_\,b\_\,a\_}$$

The walks on a torus thing



1, 4		1, 4
2, 5		5, 1
3, 5	$\Longrightarrow$	2, 5
4, 6		3, 5
5, 1		6, 3
6, 3		4, 6

Represent the lines as matrix entries. The matrix lives on a torus, so it wraps round horizontally and vertically.



Connect the entries in a cycle using the following rules

1. a column of 1s is joined vertically
2. a 1 with a vacant cell below is joined to the entry diagonally opposite the cell to its right.

The result is the ordering we want: