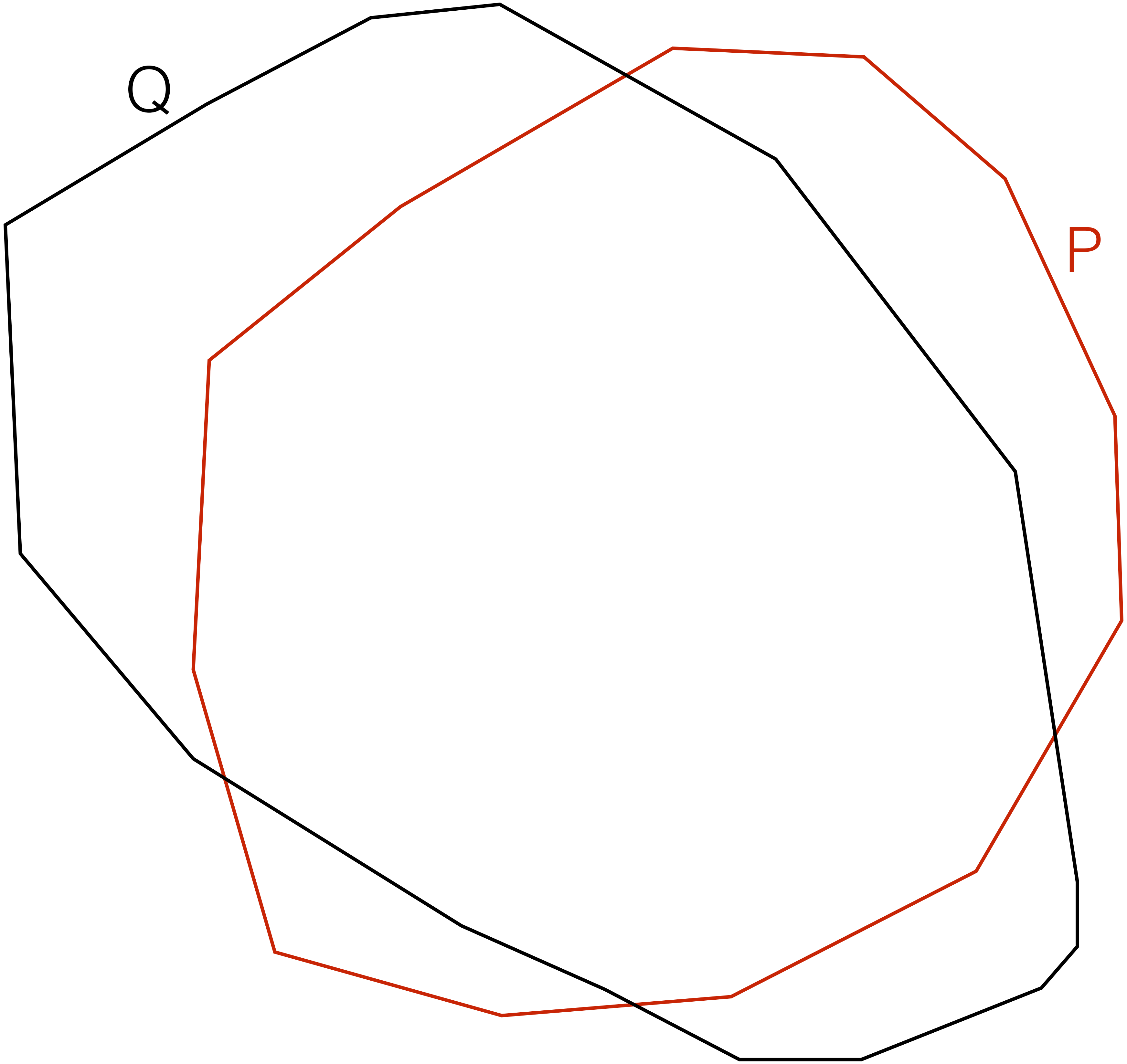
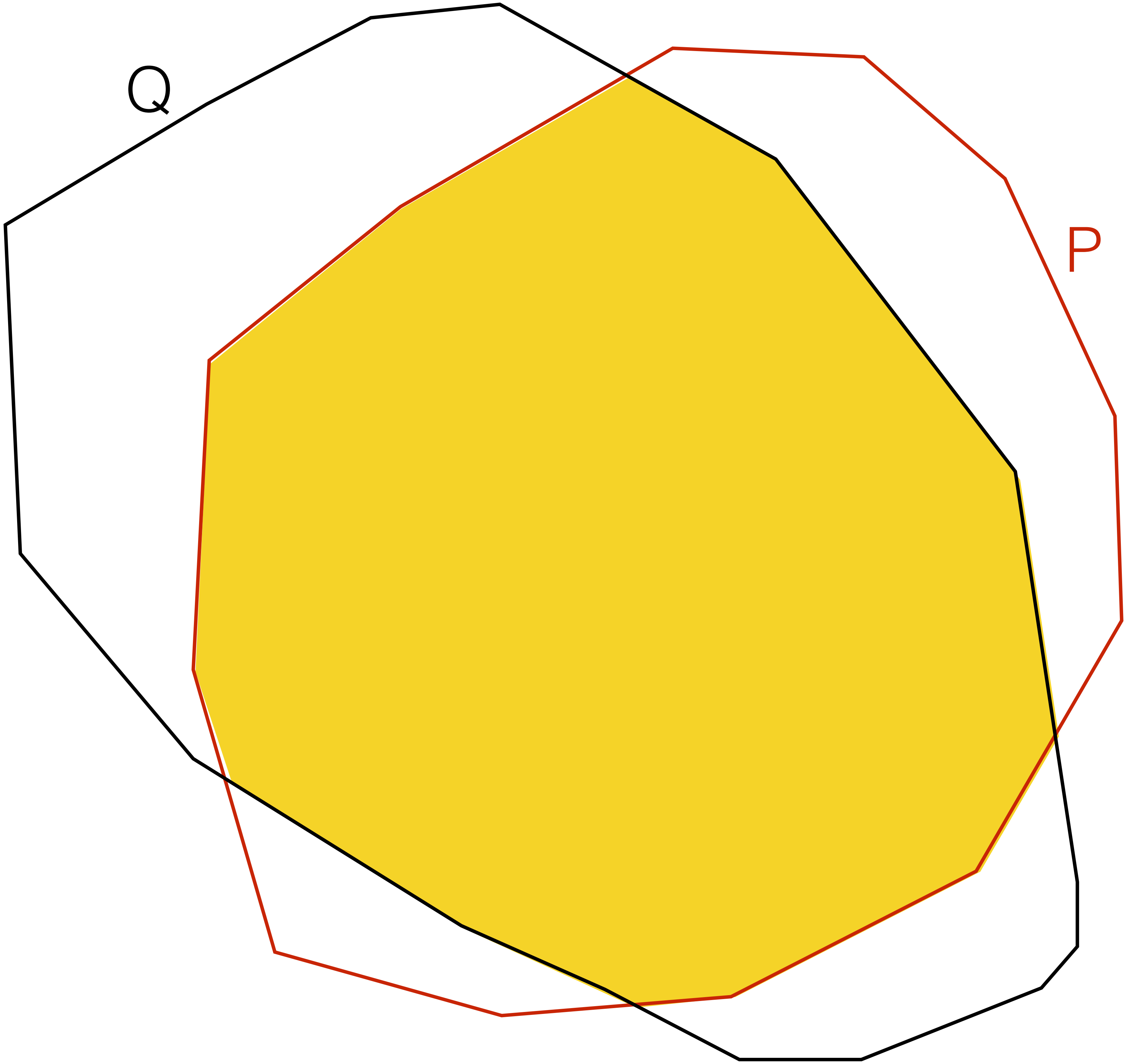
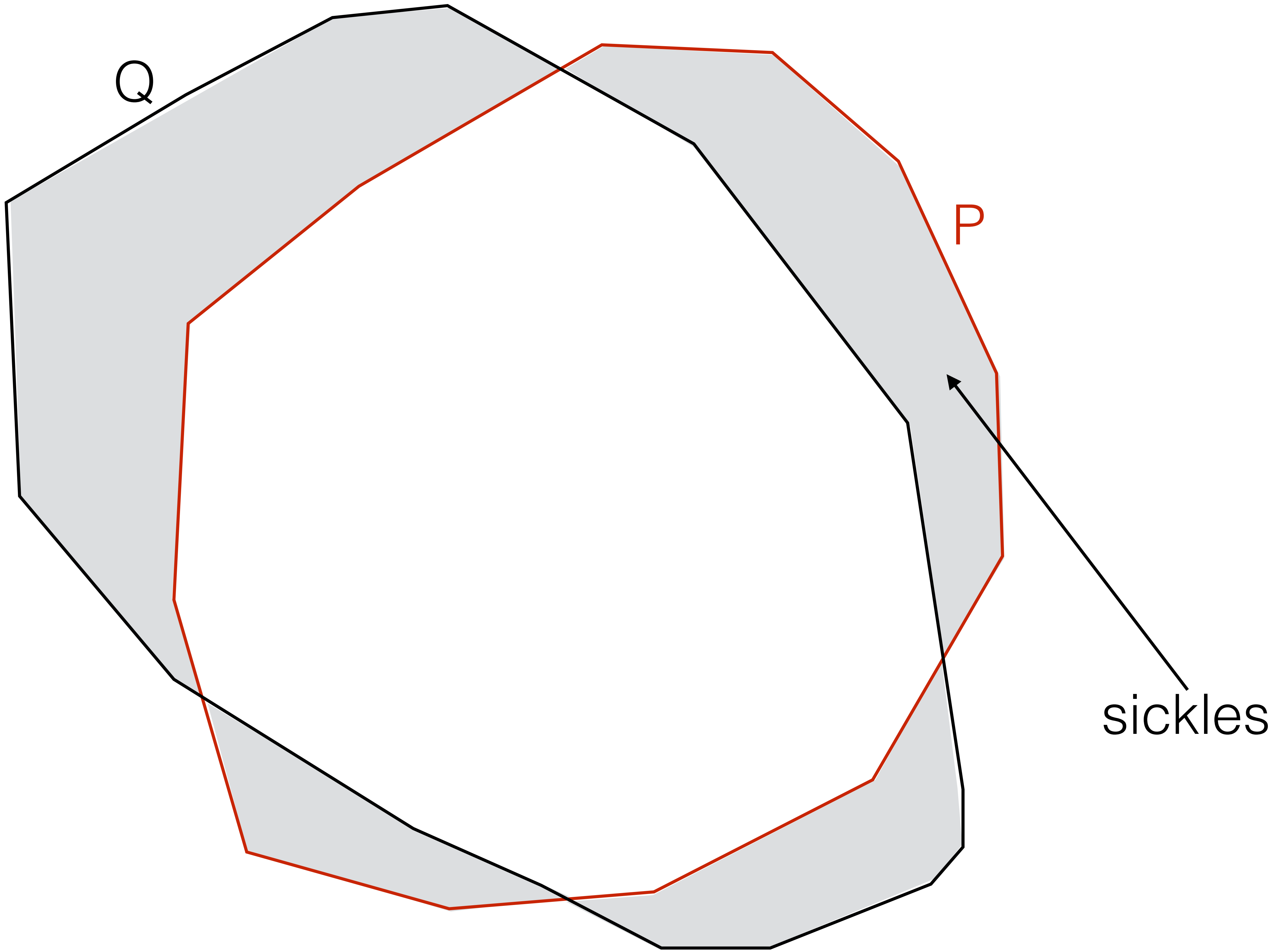
Convex polygon intersection

- The problem: Given two convex polygons, compute their intersection

- Key component in other algorithms, such as
  - computing intersection of half-planes
  - finding the kernel of a polygons
  - linear programming problems


Q
P


Q
P


Q
P
sickles

Convex polygon intersection

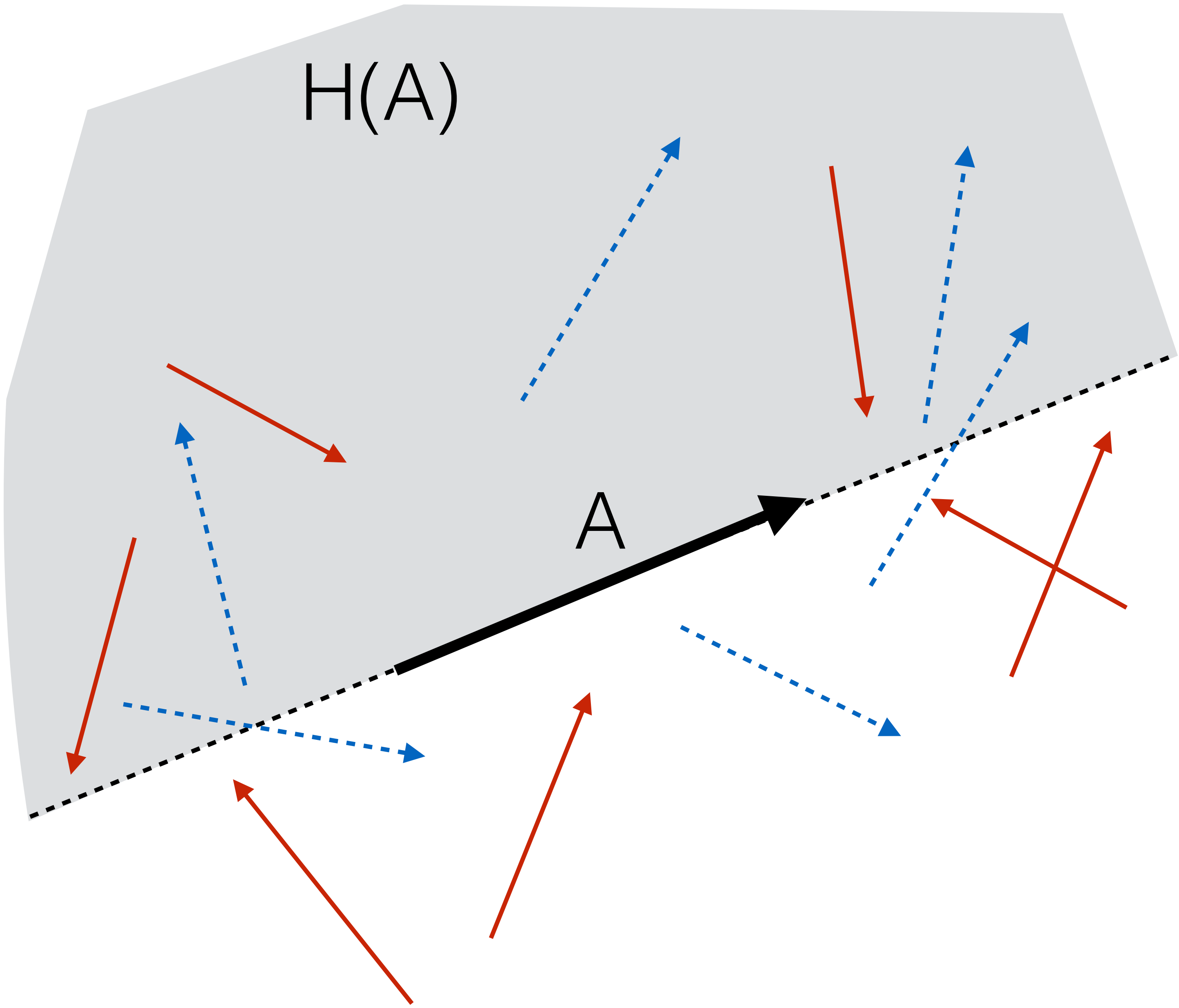
- Claim: Intersection of two convex polygons P and Q has complexity O(|P| + |Q|)
- Algorithm outline
    - choose edge A on P, B on Q arbitrarily
    - repeat
        - if A intersects B
            - print intersection (and update inside flag)
        - advance A or B
    - until both A and B cycles their polygons

Advancing

- Idea: the edges A and B chase each other, adjusting so that they meet at each intersection

Advancing

- A directed edge
- H(A): left half-plane of A



point towards A

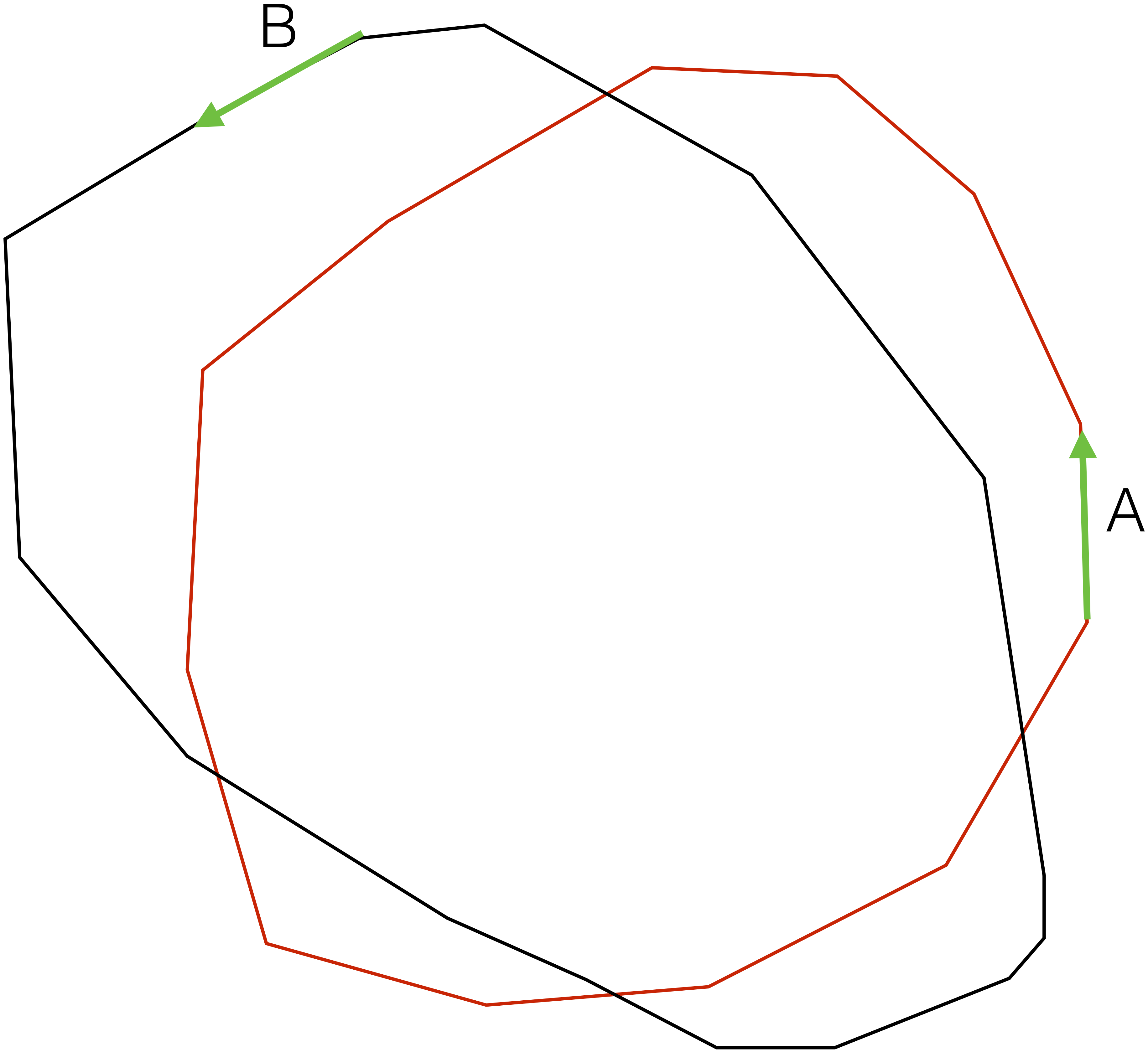
point away from A

Advancing

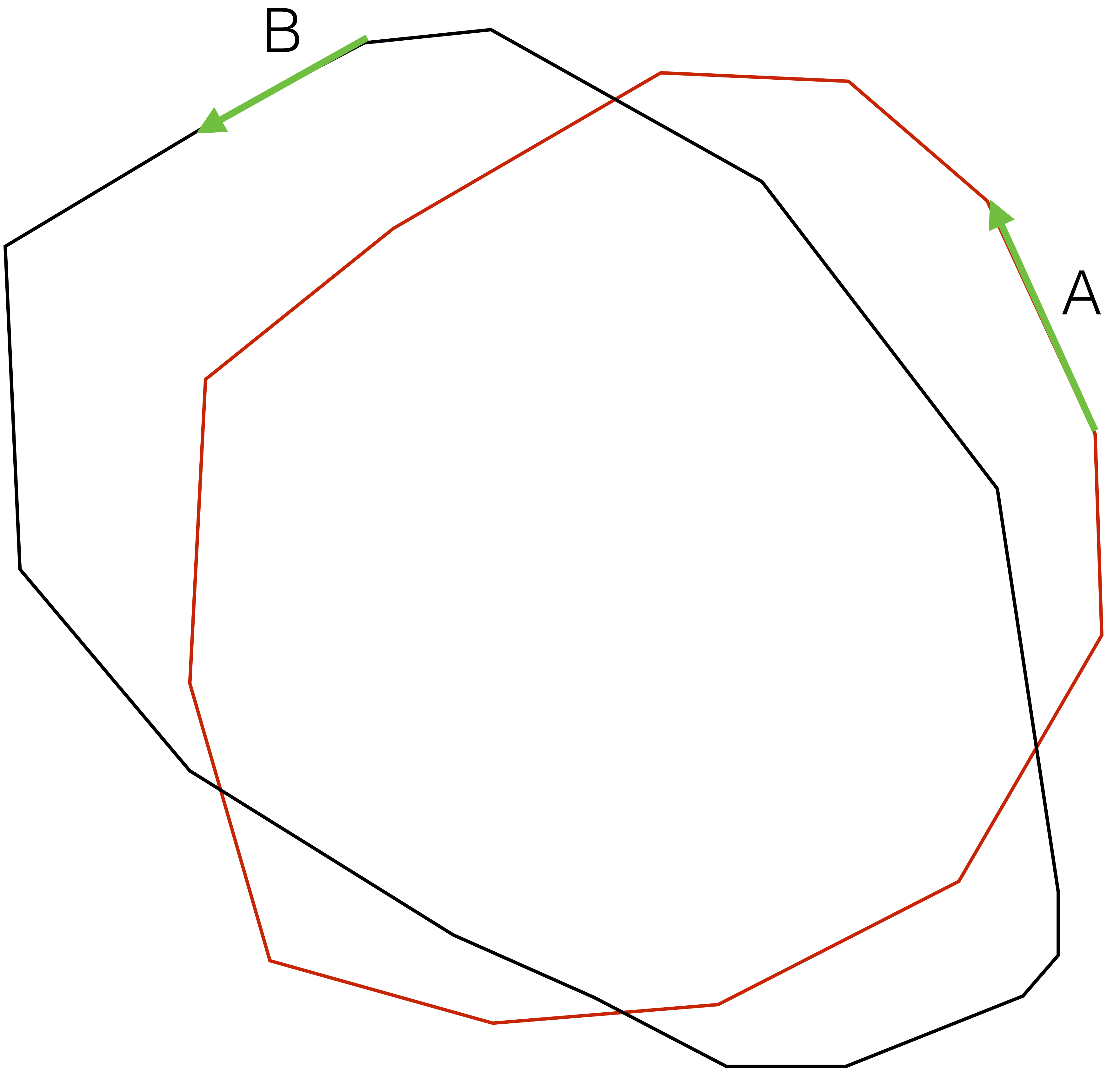
- Idea: the edges A and B chase each other, adjusting so that they meet at each intersection

- if both A and B point towards each other
  - advance whichever is outside the other
- if B points towards A and A does not point towards B
  - advance B
- if A points towards B and B does not point towards A
  - advance A
- if neither A and B point towards each other
  - advance whichever is outside the other

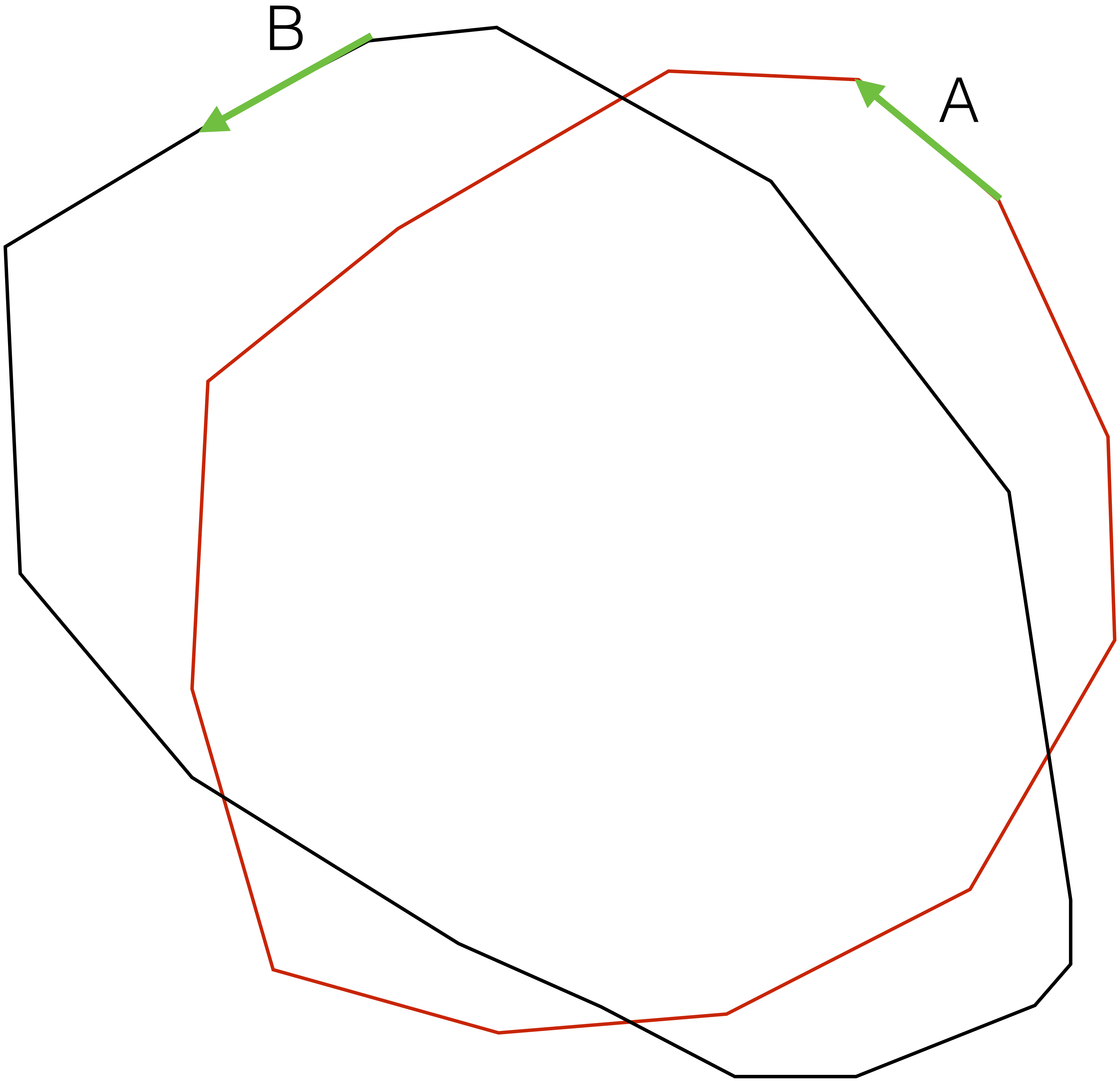
A points towards B: advance A



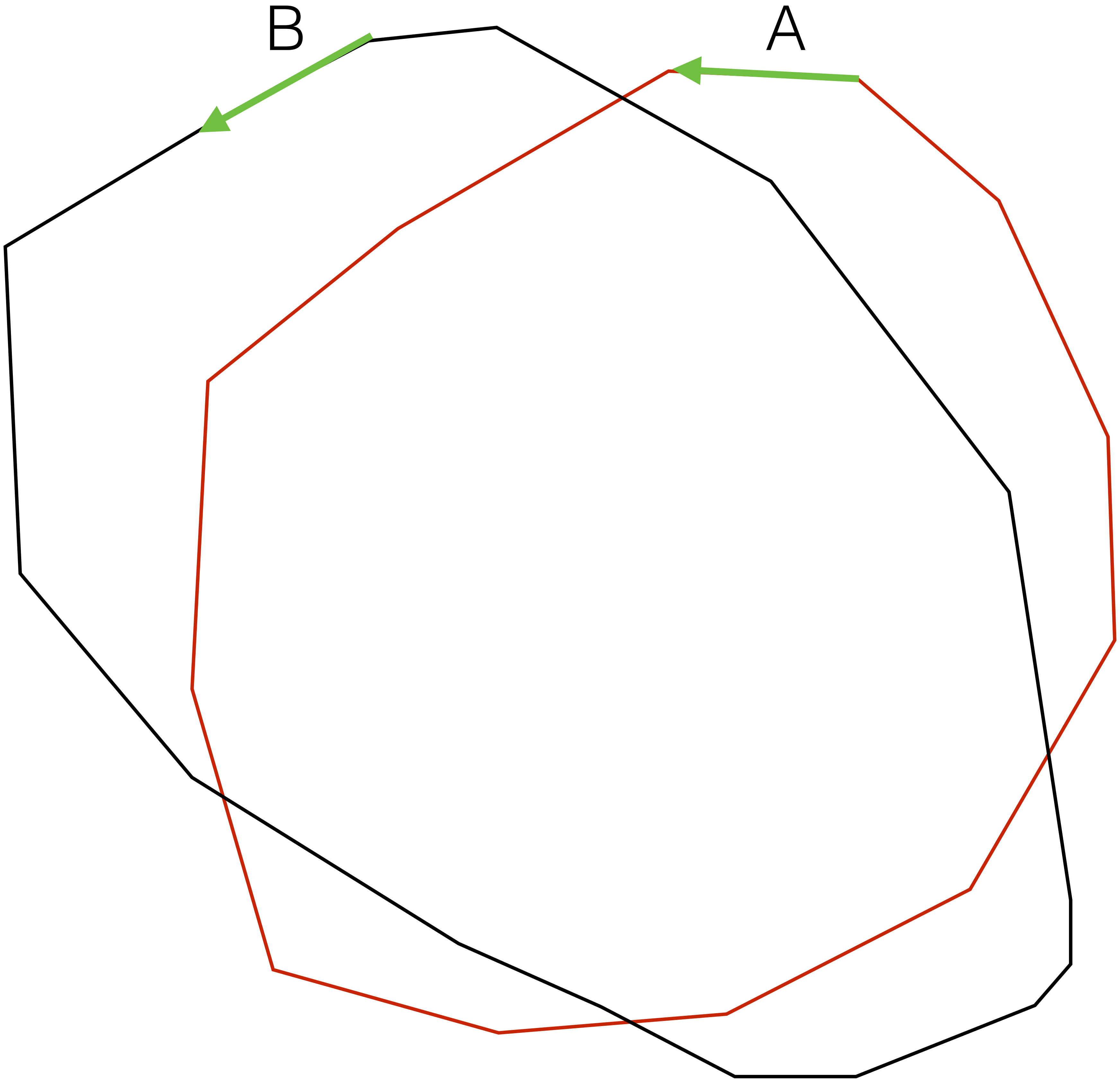
A points towards B: advance A



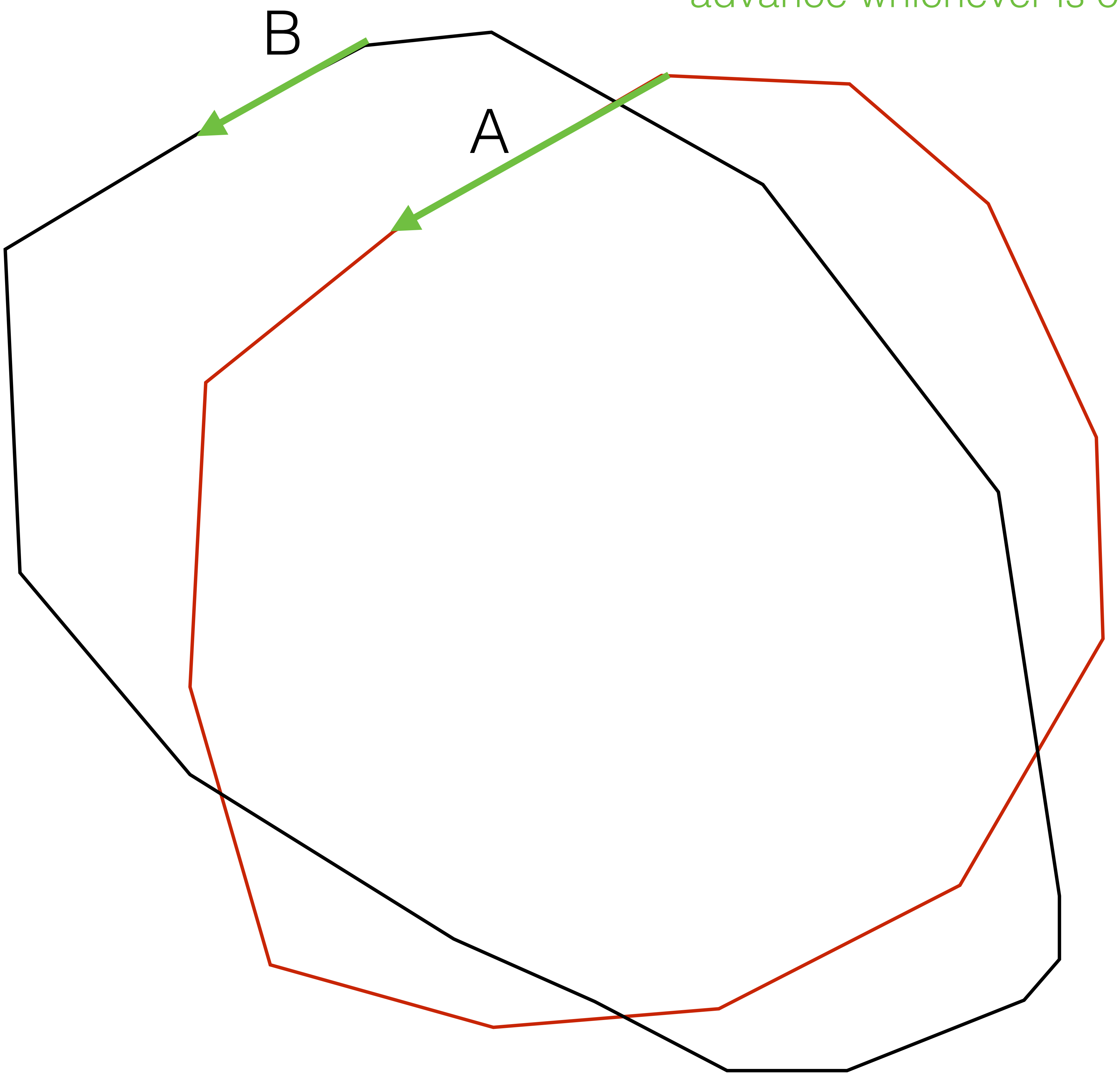
A points towards B: advance A



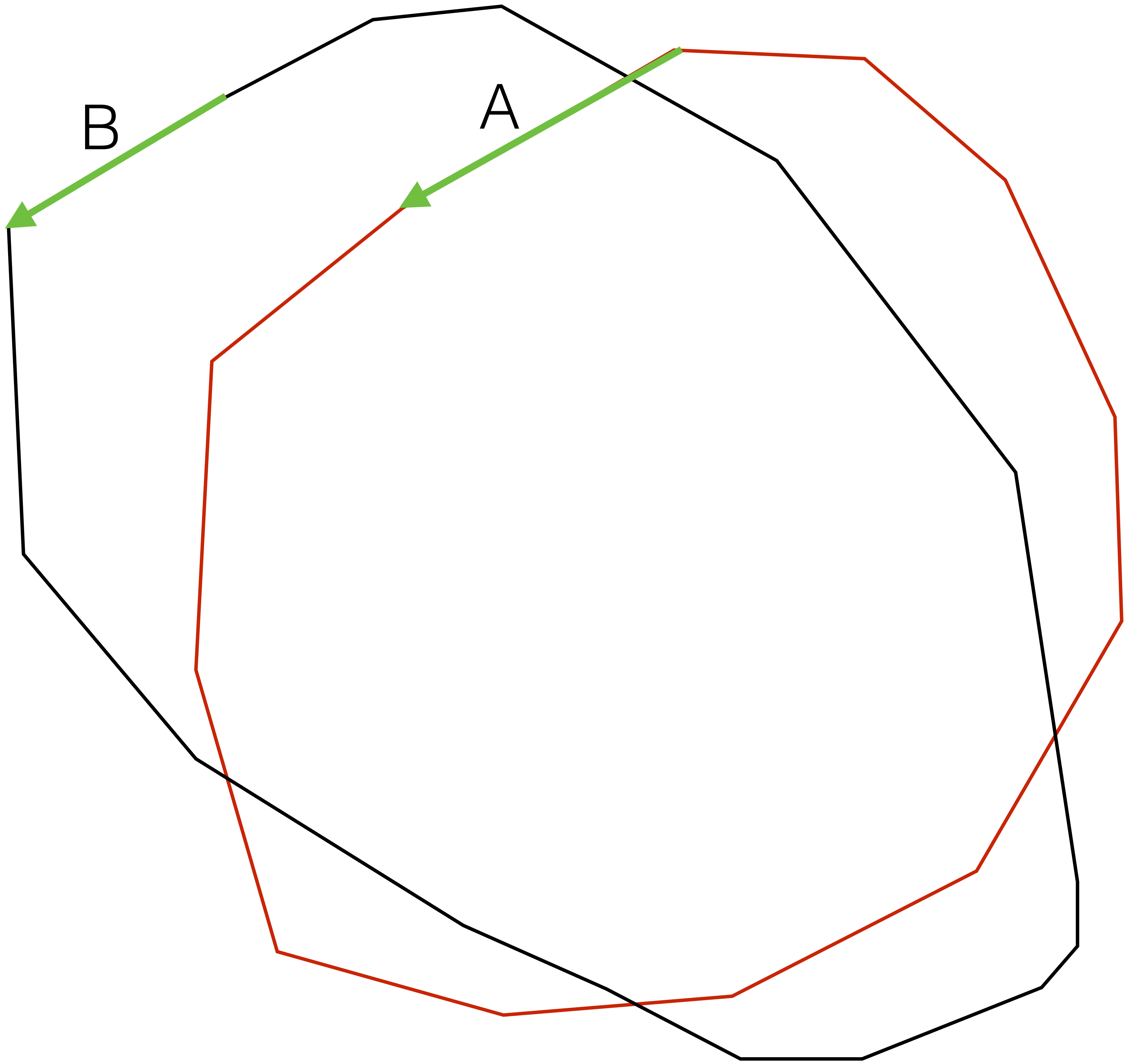
A points towards B: advance A



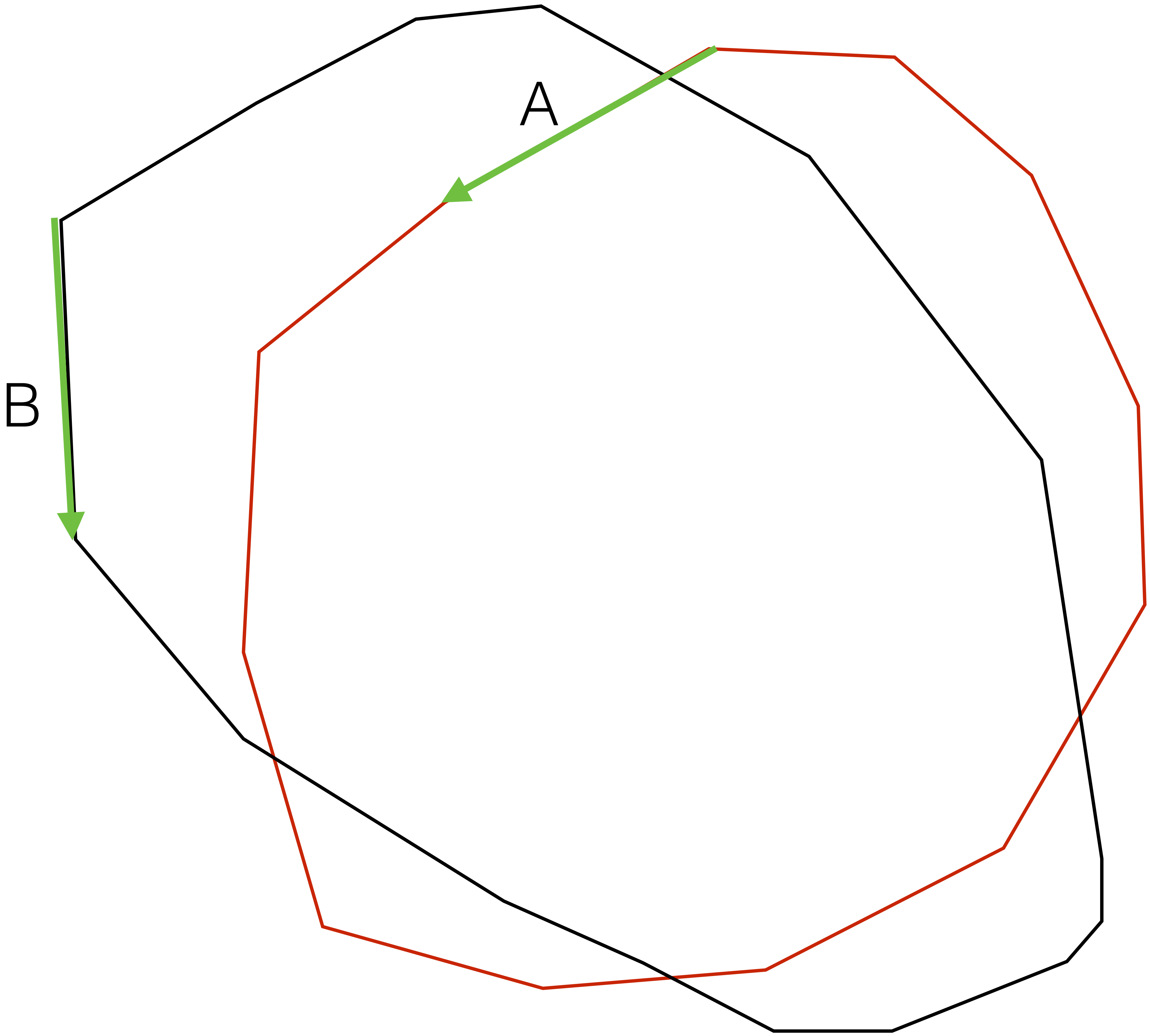
A points away from B, B points away from A:
advance whichever is outside the other



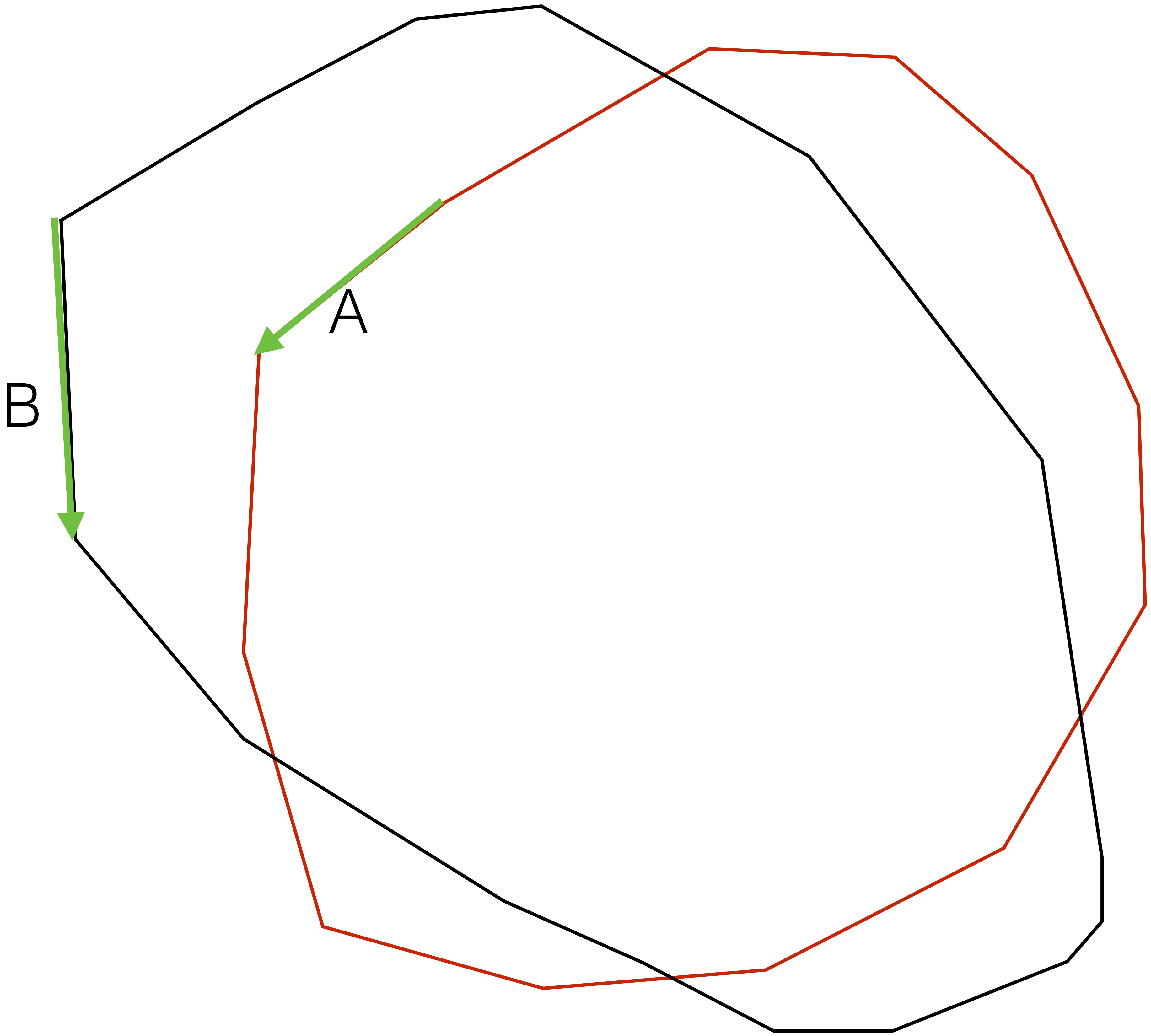
A points away from B, B points away from A:
advance whichever is outside the other



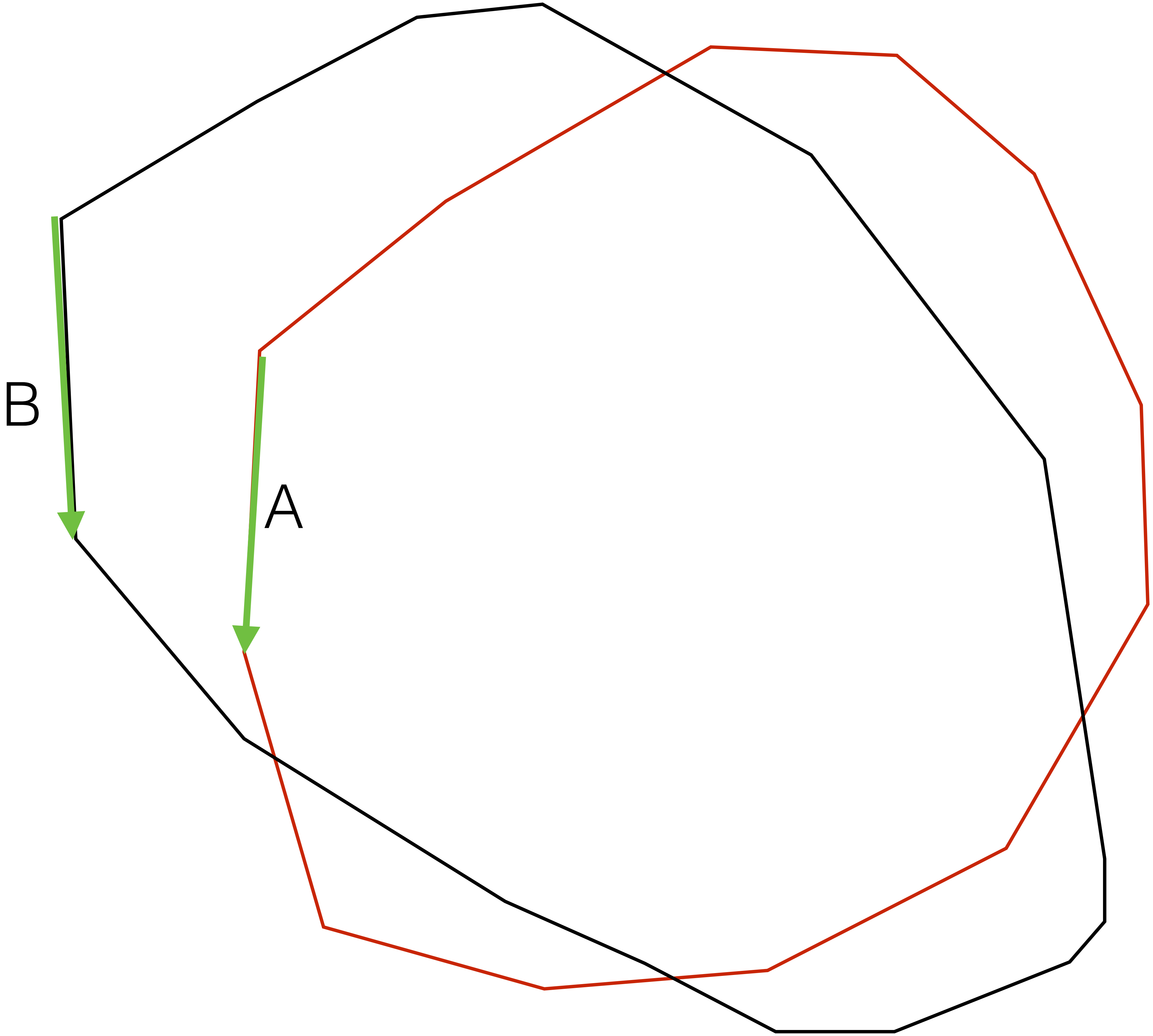
A points towards B: advance A



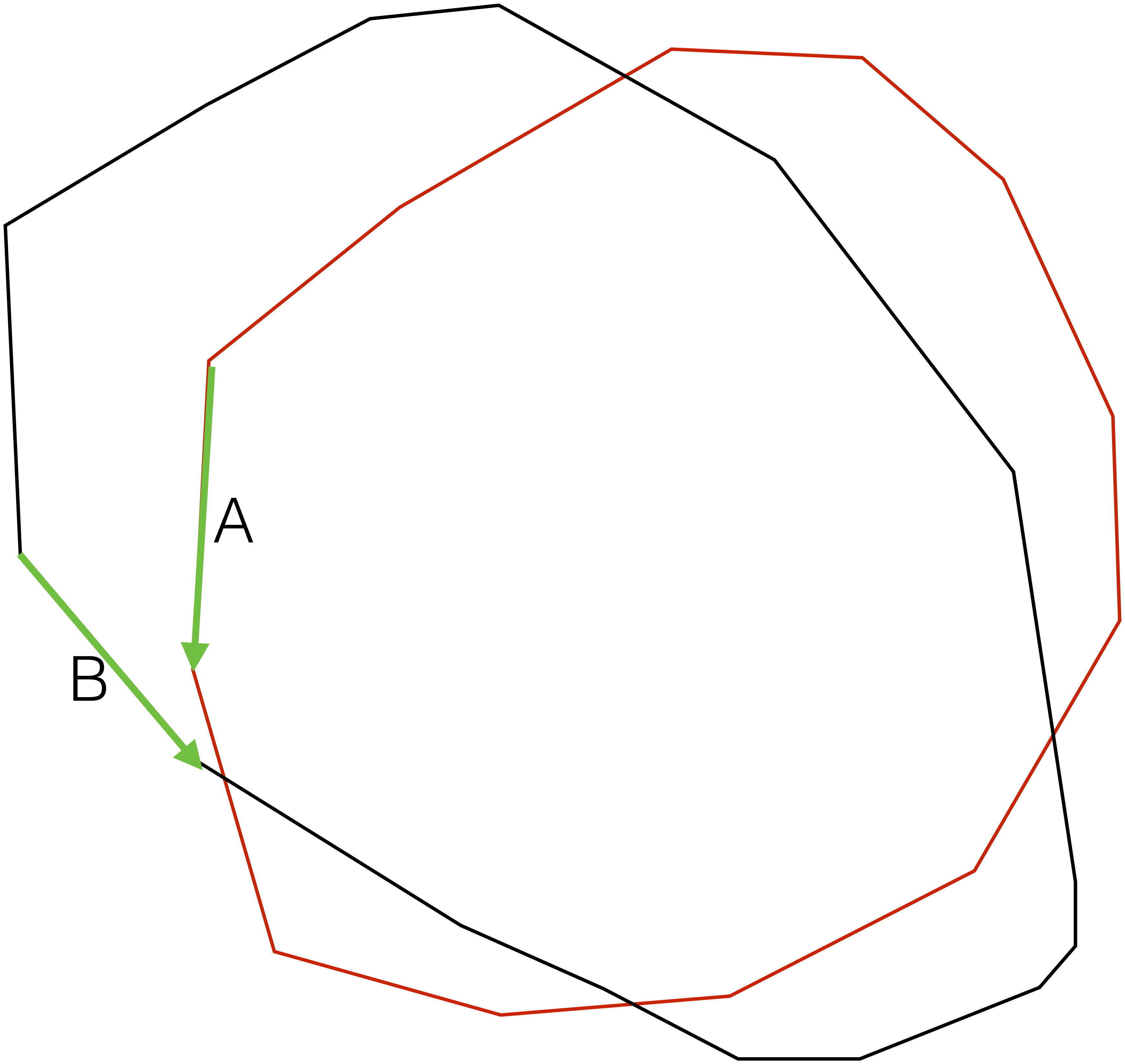
A points towards B: advance A



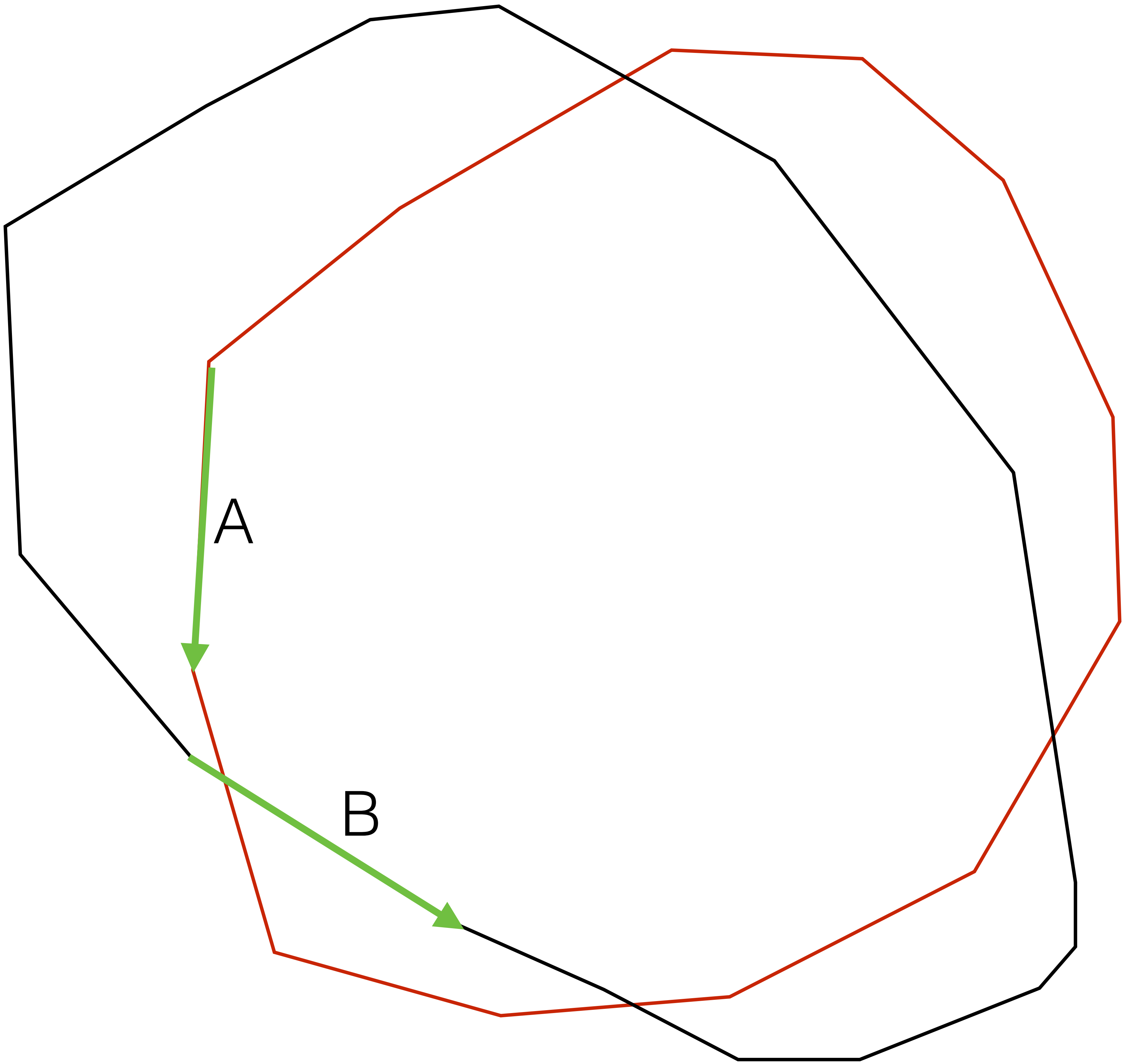
A points to B and B towards A: advance B



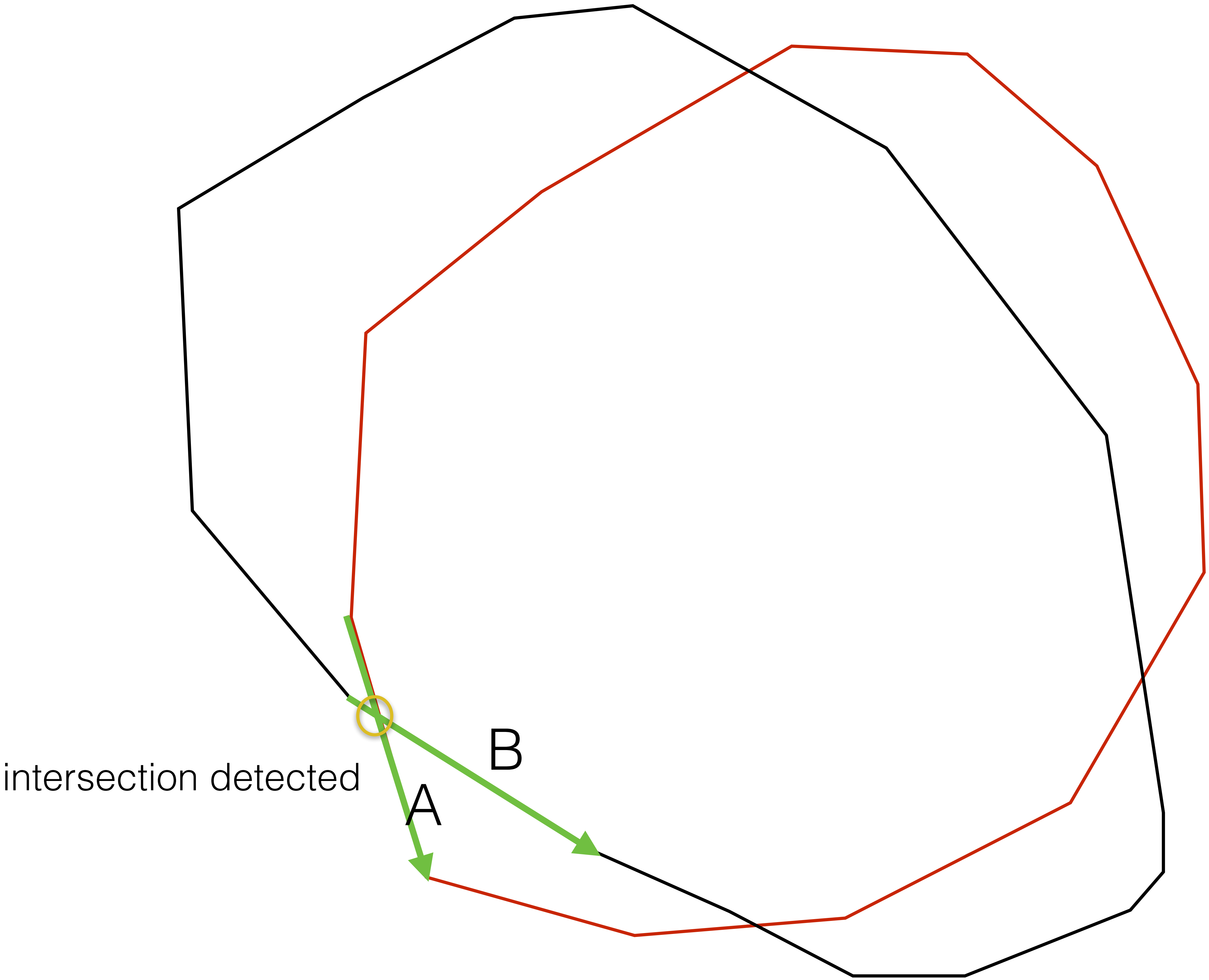
A points to B, B points to A: advance B



A points to B: advance A



B points to A: advance B



intersection detected