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 polygon
- linear programming problems

Claim: Intersection of two convex polygons \( P \) and \( Q \) has complexity \( O(|P| + |Q|) \).

Algorithm outline:
1. Choose edge \( A \) on \( P \), \( B \) on \( Q \) arbitrarily.
2. Repeat:
   - If \( A \) intersects \( B \):
     - Print intersection (and update inside flag).
     - Advance \( A \) or \( B \).
   - Until both \( A \) and \( B \) cycle their polygons.

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 $A$

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 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

intersection detected

B points to A: advance B