Competitive Programming

From Problem 2 Solution in O(1)

Computational Geometry

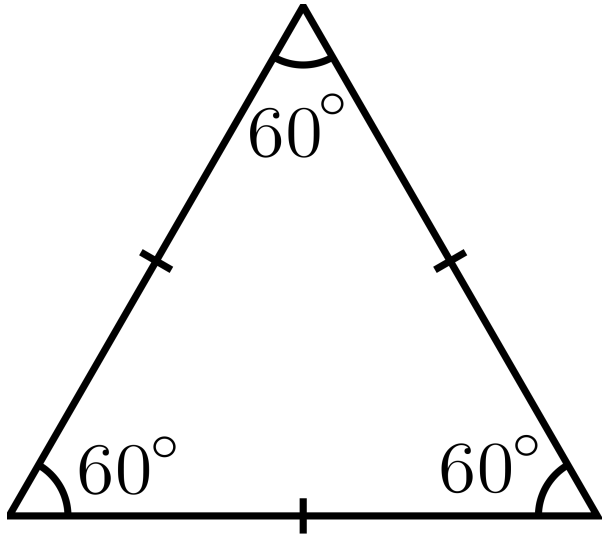
Simple and Convex Polygons

Mostafa Saad Ibrahim
PhD Student @ Simon Fraser University

Triangle

- Triangle is the **simplest polygon**
  - has 3 points
  - has 3 sides
  - has 3 angles with sum = 180
  - If all sides of equal length

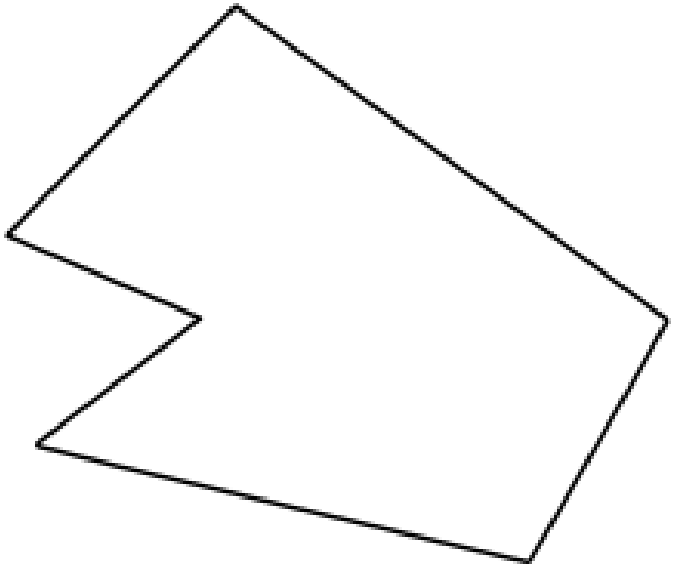
    => equal angles = 180 / 3 = 60



Src: https://upload.wikimedia.org/wikipedia/commons/thumb/9/96/Triangle.Equilateral.svg/2000px-Triangle.Equilateral.svg.png

Polygon

- Sequence of n points/vertices/corners (n >= 3)
  - Terminologies to read

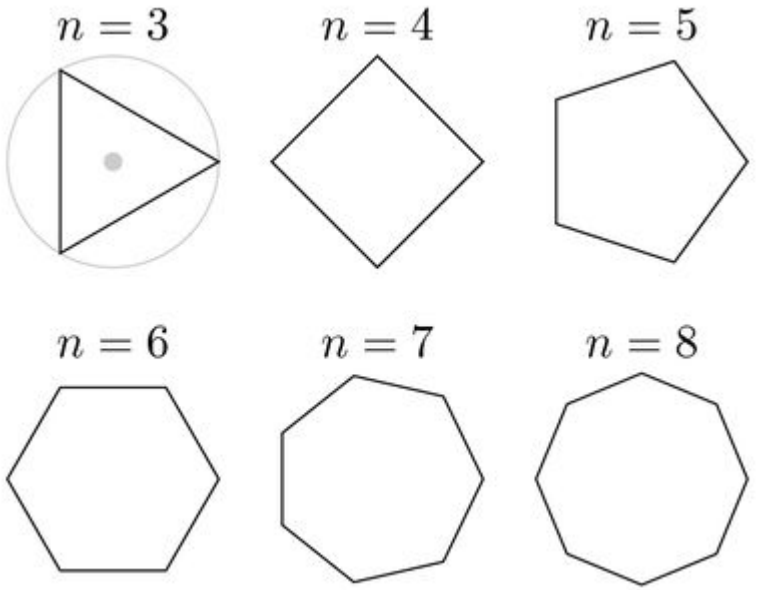


Src: http://www.basic-mathematics.com/types-of-polygons.html

Number of sides	Name of polygon
3	triangle
4	quadrilateral
5	pentagon
6	hexagon
7	heptagon
8	octagon
9	nonagon
10	decagon
12	dodecagon

Regular Polygon

- It is equiangular and equilateral
  - Equal Angles, Equal Sides
- Internal angle =
  (n-2) * 180 / n



Src: http://www.mathguide.com/lessons/RegularPoly/regularpoly.gif

Polygon: Implementation

- Typically polygon of n points is represented with vector of n points
- Then (p[i], p[i+1]) is an edge: i => [0 - n-1]
- Last edge is problematic
  - One way use: (p[i], p[(i+1)%n]) to wrap to begin
  - Note: Mod is expensive...and code wise sometimes annoying
  - Other trick: add p[0] to the vector of points (n+1 points)
  - Iterate till n exactly (not p.size())
  - From problem to another, determine what fits more

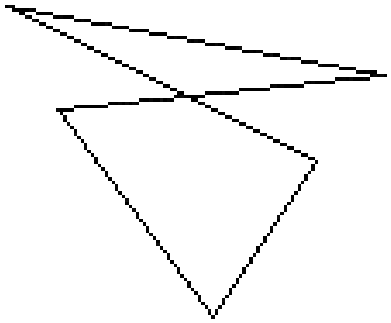
Simple Polygon

- A flat shape consisting of straight, **non-intersecting** line segments

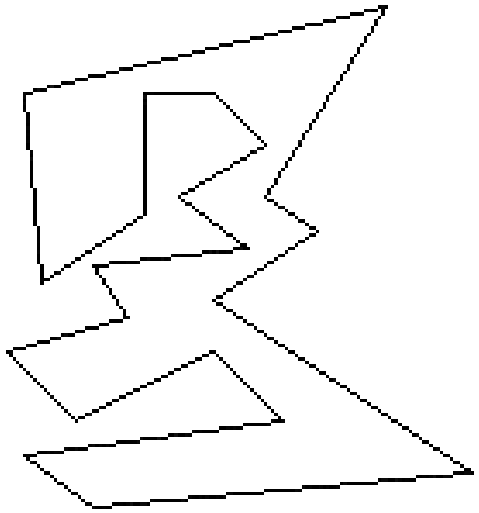
Src: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3c/Polygons_Examples_of_polygons.png/220px-Polygons_Examples_of_polygons.png

Non Simple Polygon

- Intersecting edges



Not a Simple Polygon



Simple Polygon

Src: http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/Definitions/definition.htm

Is Simple Polygon Implementation

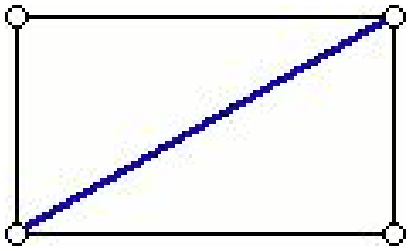
- BruteForcing it is easy in $O(n^2)$
  - just make sure no 2 edges intersects
  - Note 1: Don't check consecutive edges
  - Note 2: Remember last edge, edge 1 are consecutive
  - Use ccw/cp to test intersection
- Other advanced ways
  - O(nlogn) using line sweep
  - O(n) using very complex algorithm

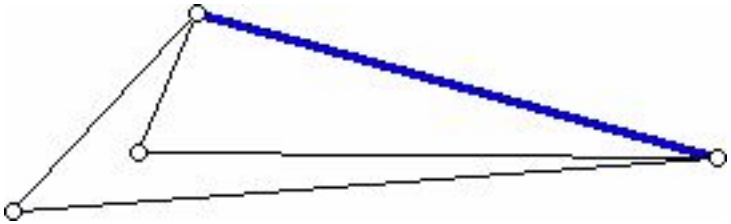
Simple Polygon Implementation

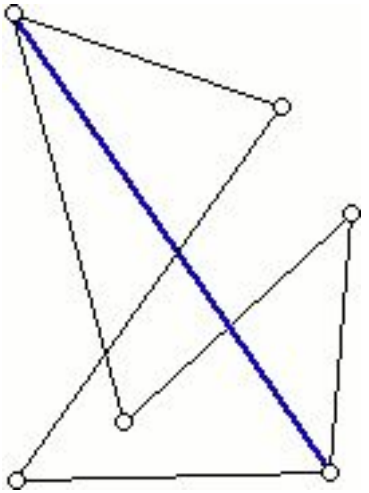
```
// assume p0 is added again to p (n+1 points)
bool isSimplePolygon(vector<point> &p) {
  for (int i = 0; i < sz(p) - 1; i++) {
    for (int j = i + 2; j < sz(p) - 1; j++)
      if (intersect(p[i], p[i + 1], p[j], p[j + 1])
          && (i != 0 || j != sz(p) - 2) /* last/first edges are consecutive*/)
        return false;   //Segments must not share vertices
  }
  return true;
}
```

Polygon Diagonals

- A polygon's diagonals are line segments from one vertex (point) to another, but not the sides.
- The number of diagonals of an n-sided polygon is: $n(n-3)/2$



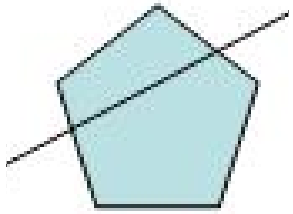




Src: http://www.ams.org/samplings/feature-column/fcarc-diagonals2

Convex Polygon

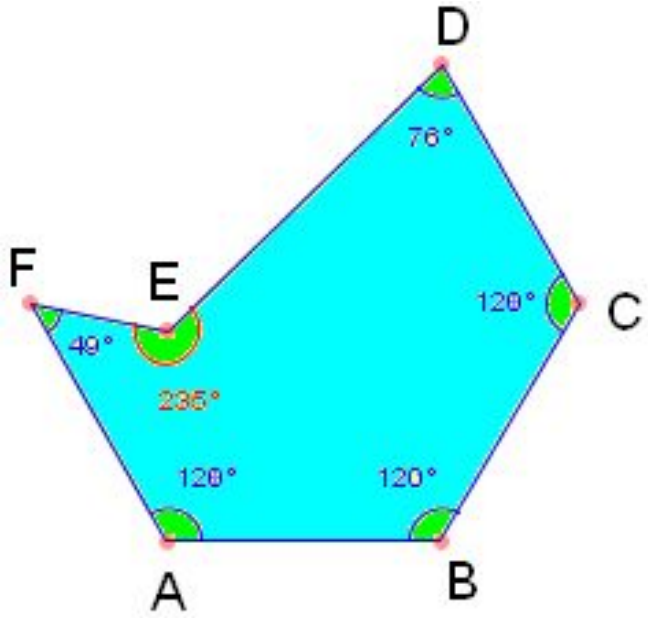
- All interior angles <= 180
  - E.g. all the vertices of the polygon will point outwards, away from the interior of the shape
- A line drawn through a convex polygon will intersect the polygon exactly twice
- All the diagonals of a convex polygon lie entirely inside the polygon

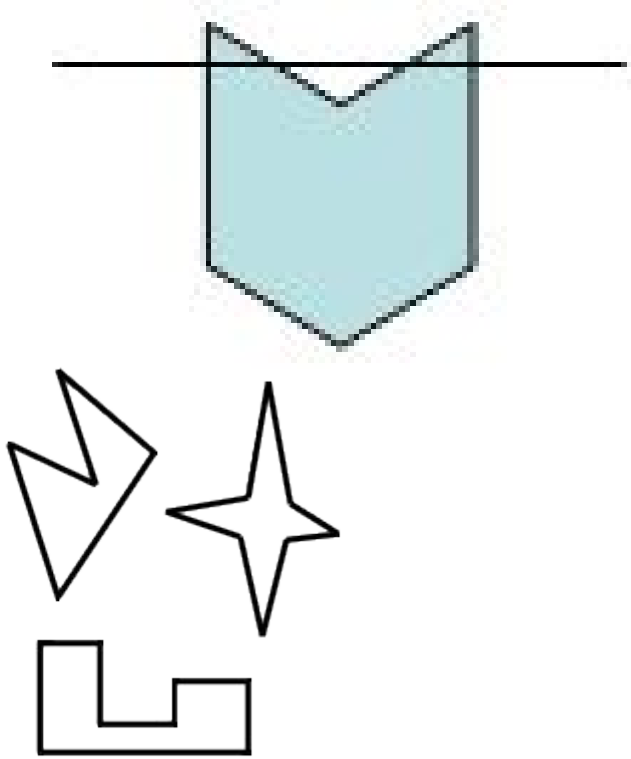


Src: http://www.mathopenref.com/polygonconvex.html

Concave Polygon

- Non convex one





Src: http://image.tutorvista.com/cms/images/38/concave-polygon.jpg

Is Convex Polygon Implementation

- One can just iterate / compute each angle
  - if all <= 180 => convex
  - This involves double computations
- What about using CCW test?
- Let's assume the polygon points are ordered
  - E.g. all are in ccw order or cw
- If every 2 edges have same direction (e.g. all are ccw), then this is convex
  - If there mix of signs, this is concave

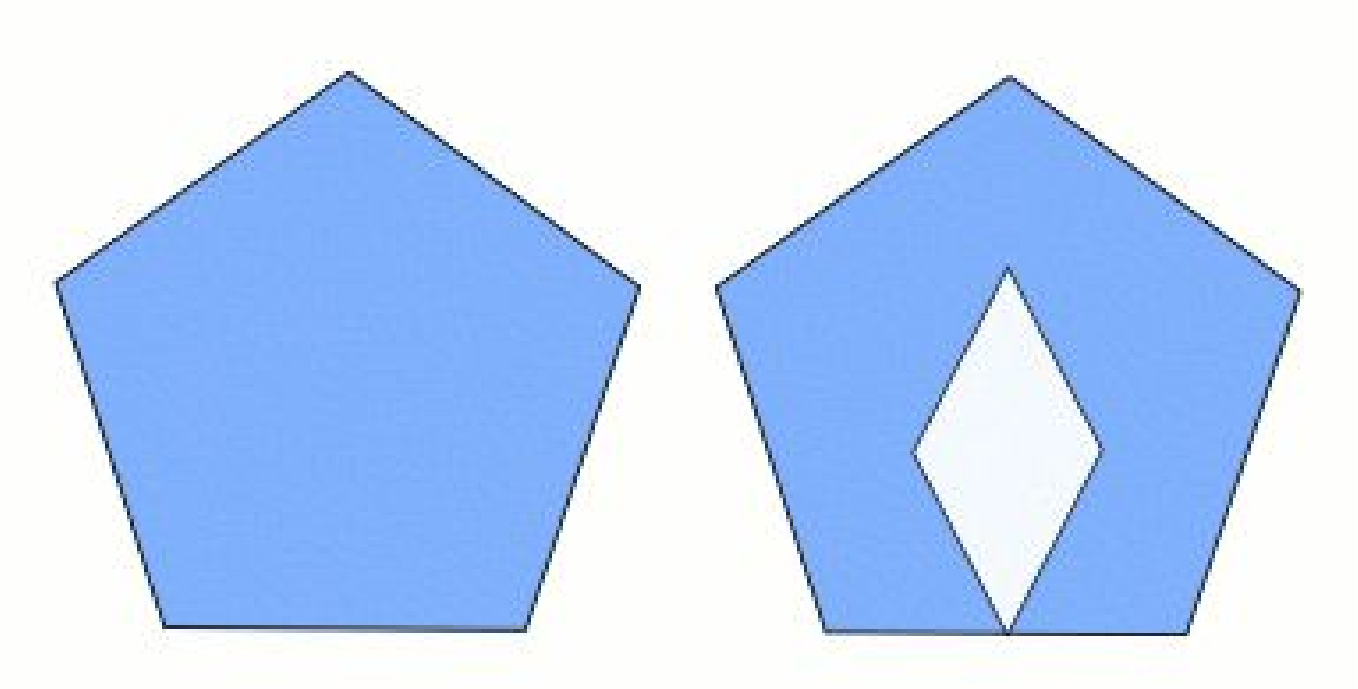
Convex Polygon Implementation

```
bool isConvexPolygon(vector<point> &p) {
  // all polygon 3 consecutive points must have same sign (ccw or cw)
  p.push_back(p[0]), p.push_back(p[1]); // wrap points for simplicity

  int sign = ccw(p[0], p[1], p[2]);
  bool ok = true;
  for (int i = 1; ok && i < sz(p) - 2; i++) {
    if (ccw(p[i], p[i + 1], p[i + 2]) != sign)
      ok = false;
  }
  p.pop_back(), p.pop_back();
  return ok;
}

// isConcave = !isConvex
// isConcave = 2 different signs
```

Polygon with holes (ignore)



Src: https://plus.maths.org/issue43/features/kirk/polygon_hole.gif

Critical Notes

- Polygon input
  - **Coordinates** are integer or real?
  - Points are **ordered**? yes: ccw? cw? on angle with P0?
  - Caution: Sometimes input has **duplicate** points
  - Caution: Sometimes input has **collinear** points
  - Caution: for invalid input (n < 3), special cases: n = 3
- Polygon type
  - Remember keywords: **simple/!simple - convex/concave**
  - Some algorithms for simple and/or convex polygons are very fast than generic case

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً