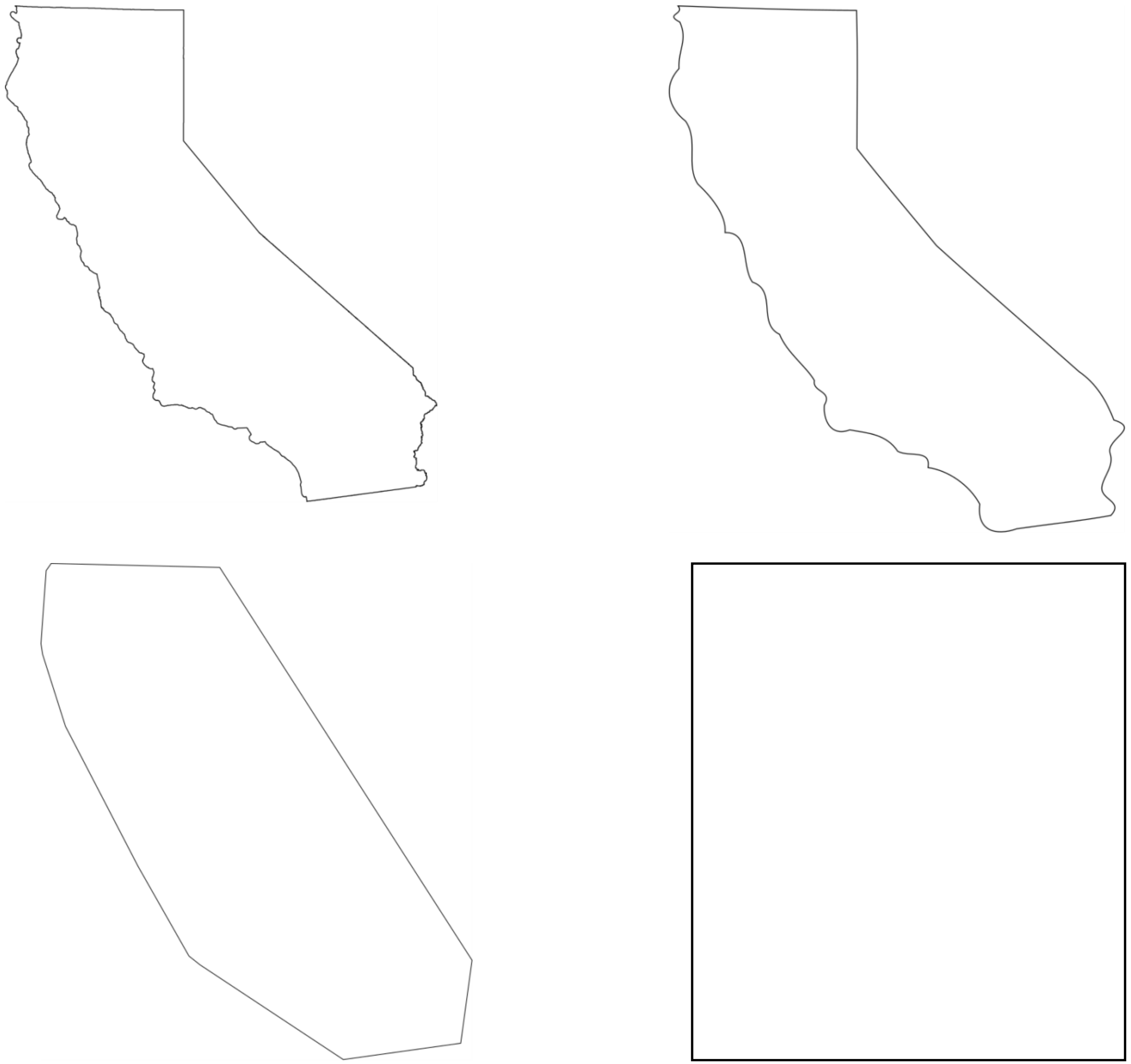
CS133
Computational Geometry

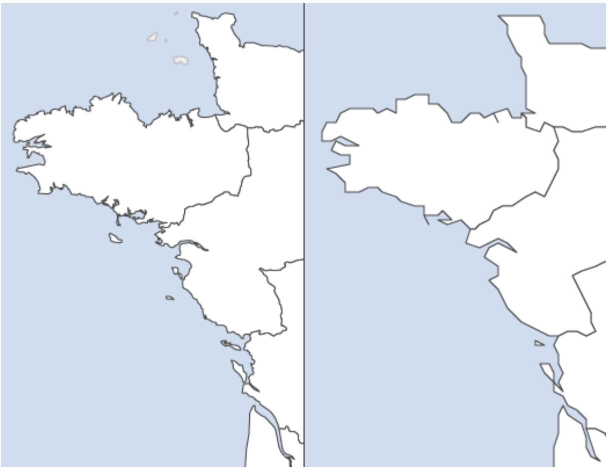
Simplification Algorithms

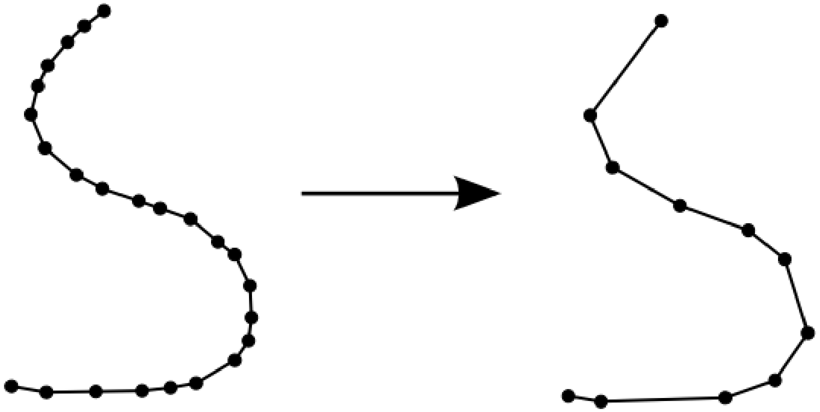
Line Simplification



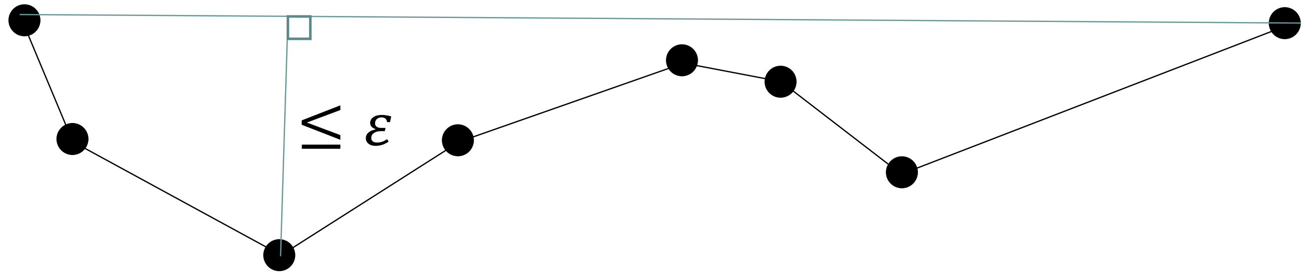
Line Simplification

- Given a line string and a distance threshold ε, return a simplified line string such that any point in the input line string is not displaced more than ε

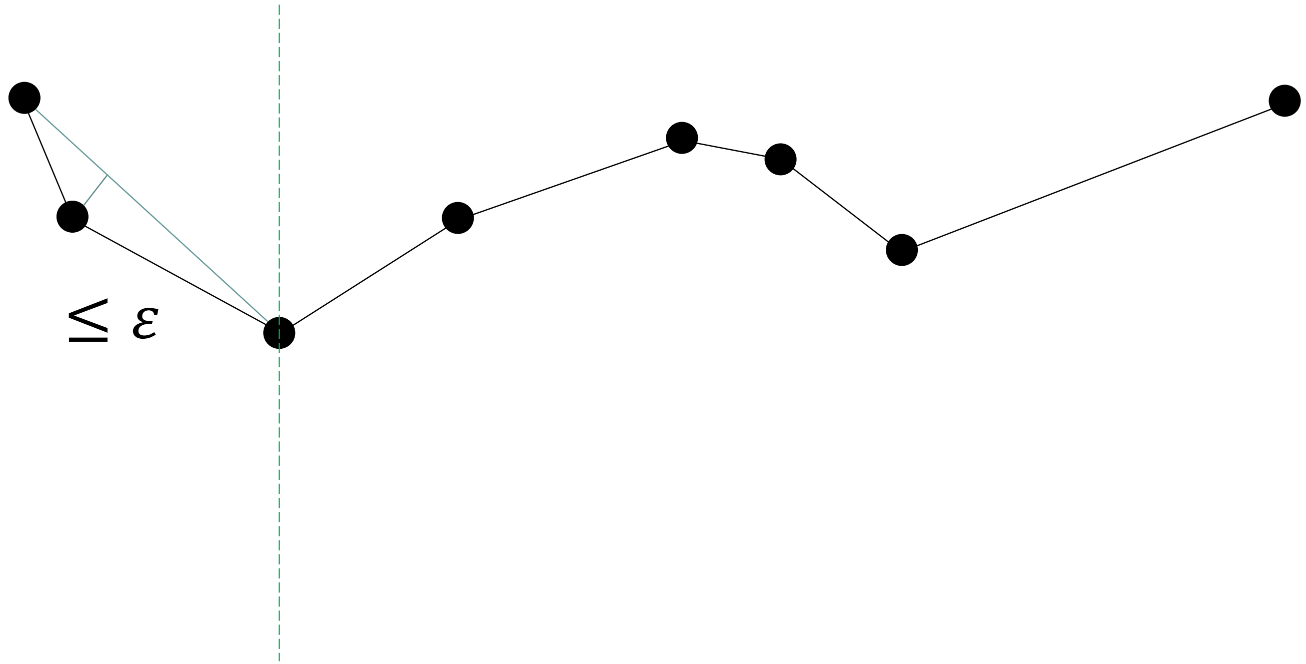




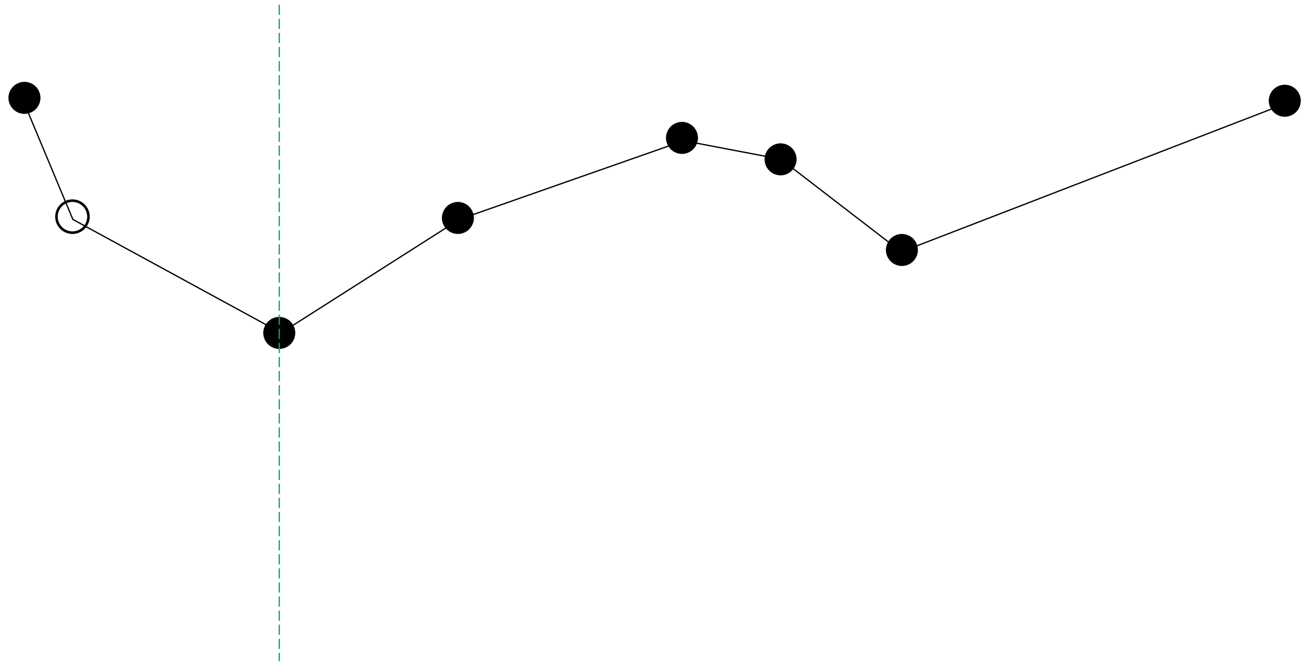
Douglas–Peucker Algorithm



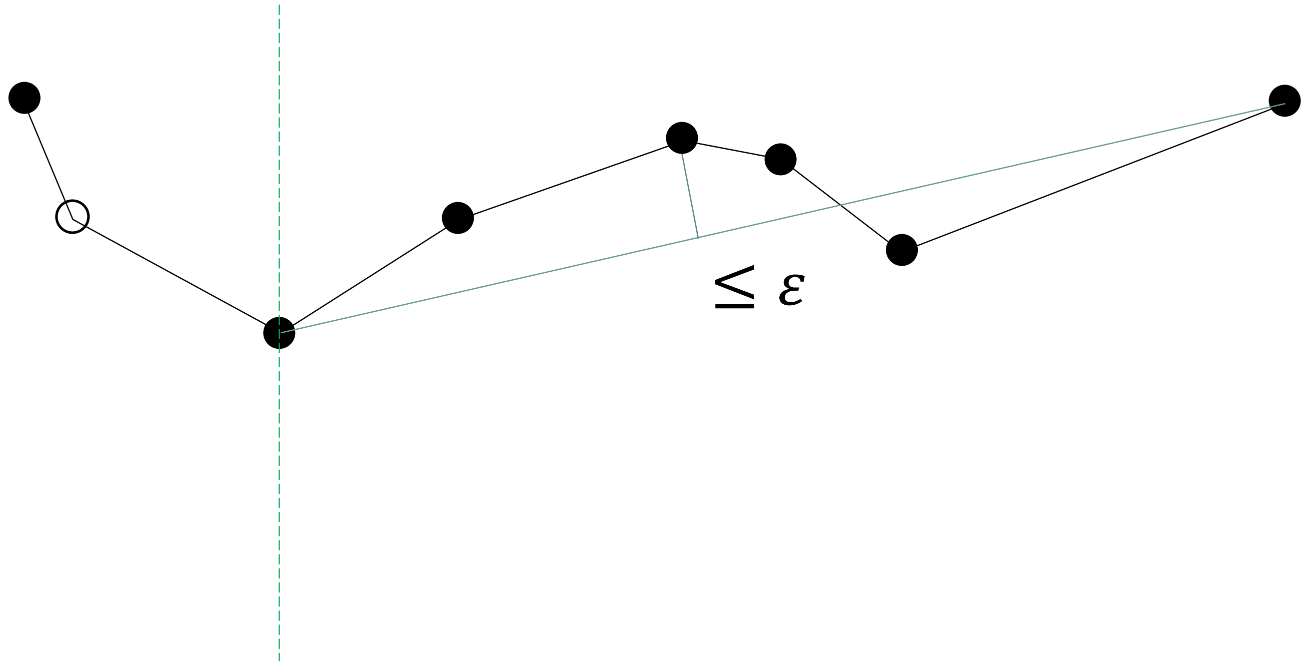
Douglas–Peucker Algorithm



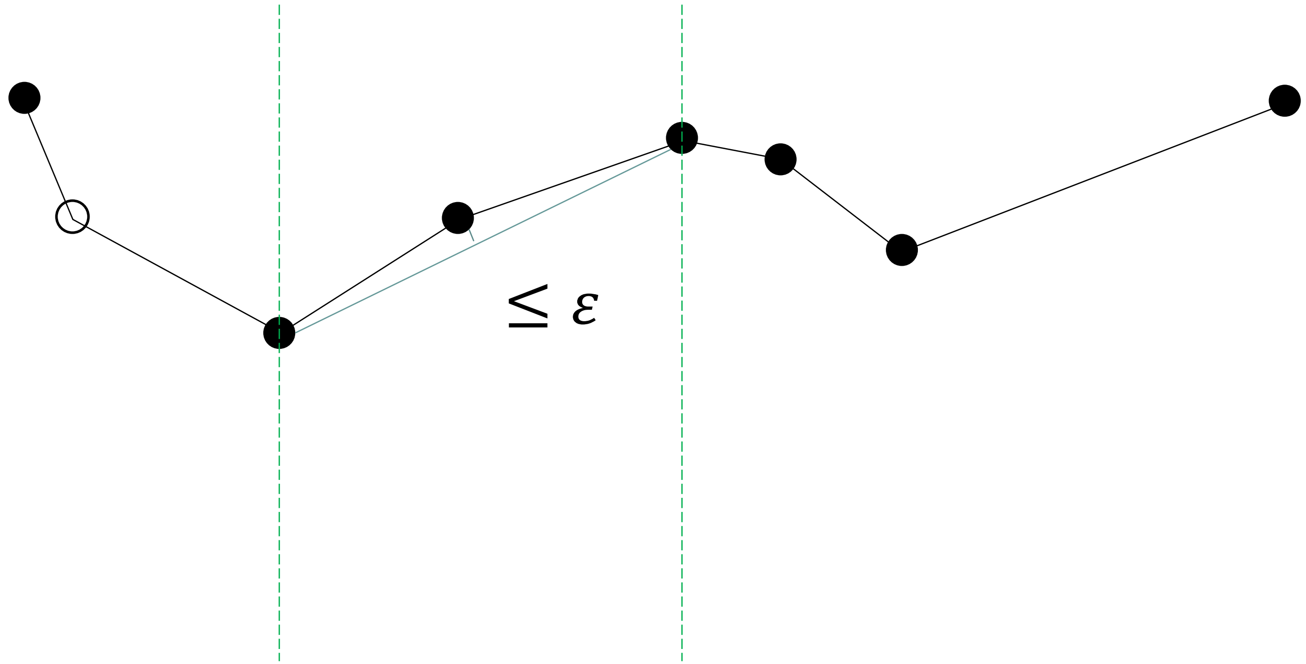
Douglas–Peucker Algorithm



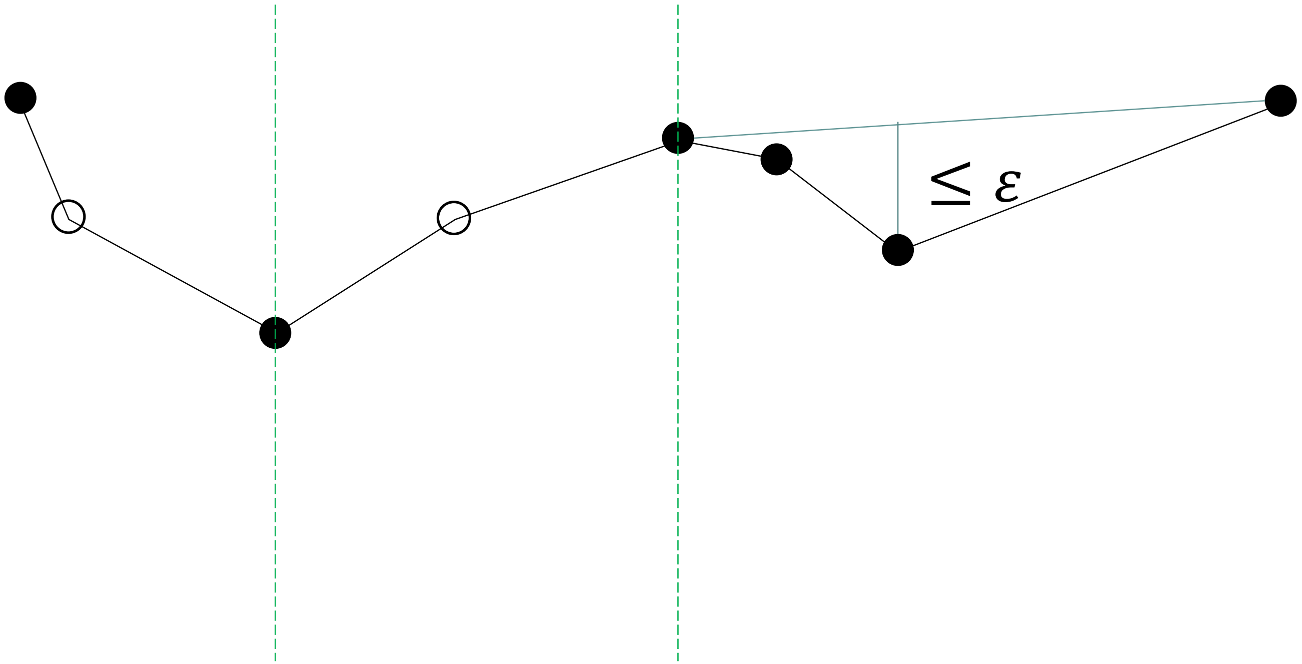
Douglas–Peucker Algorithm



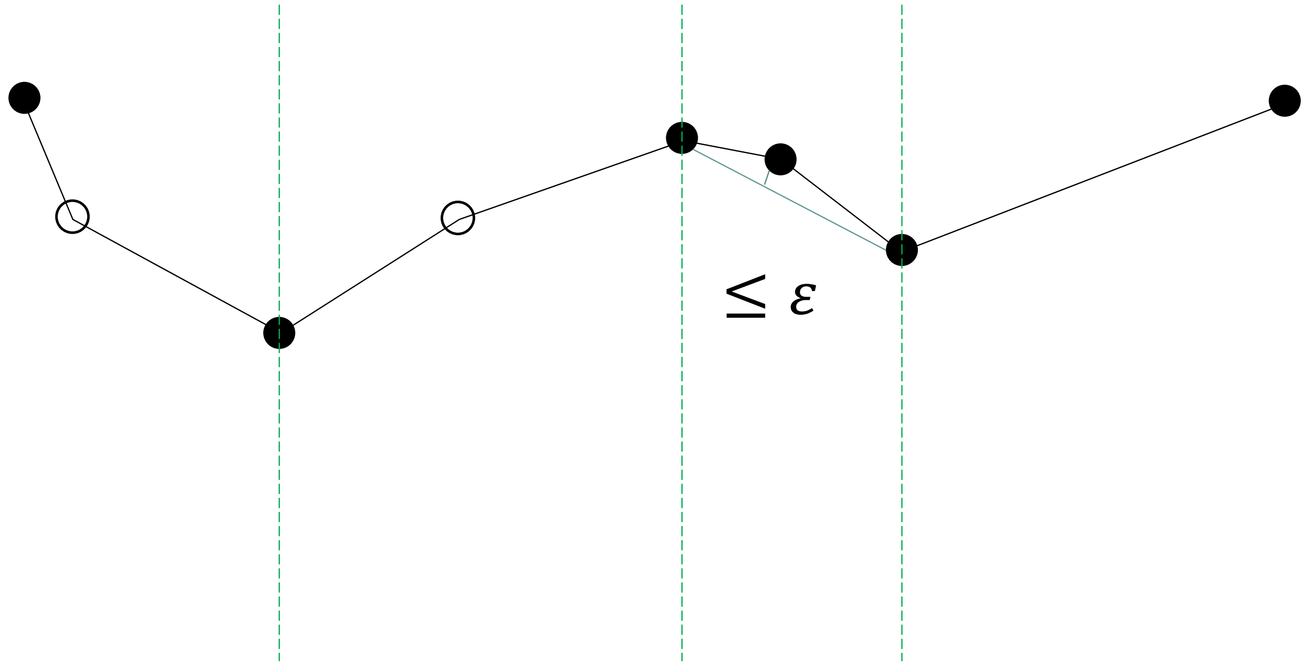
Douglas–Peucker Algorithm



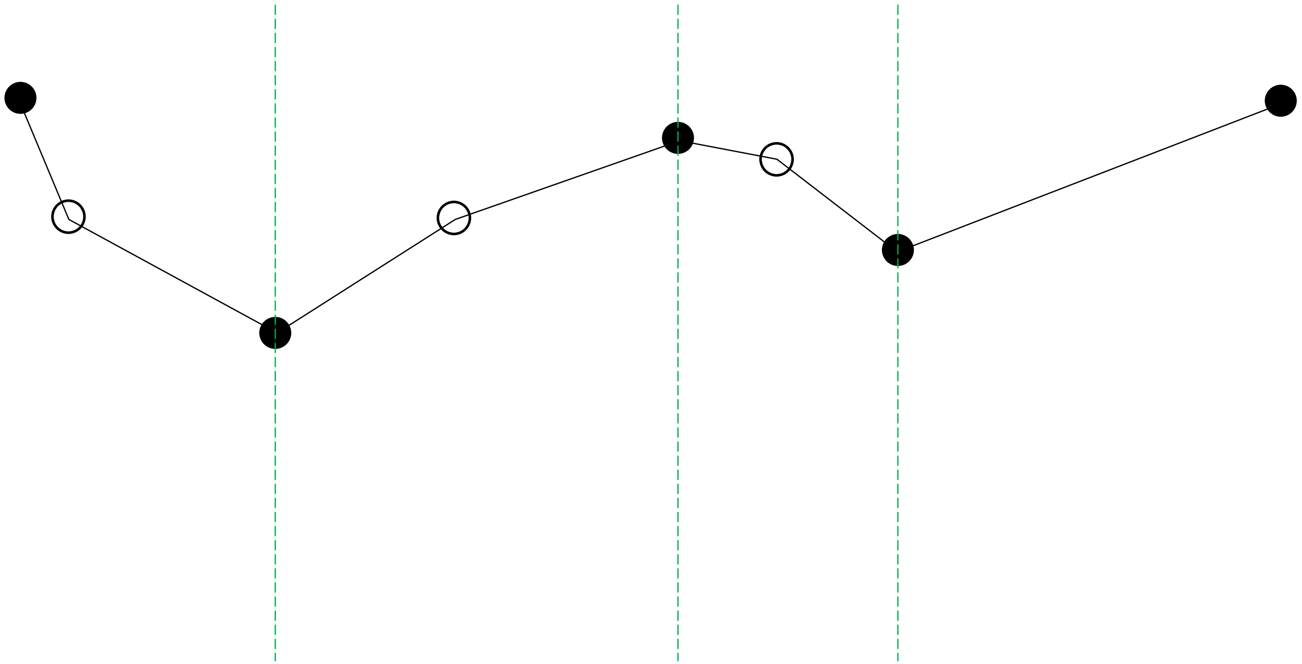
Douglas–Peucker Algorithm



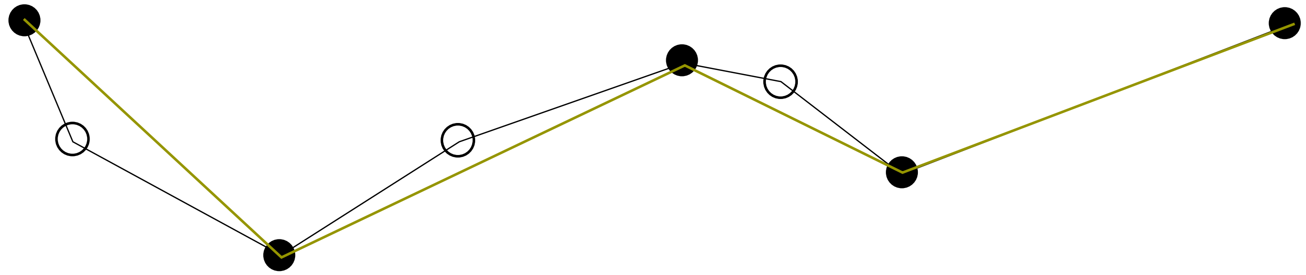
Douglas–Peucker Algorithm



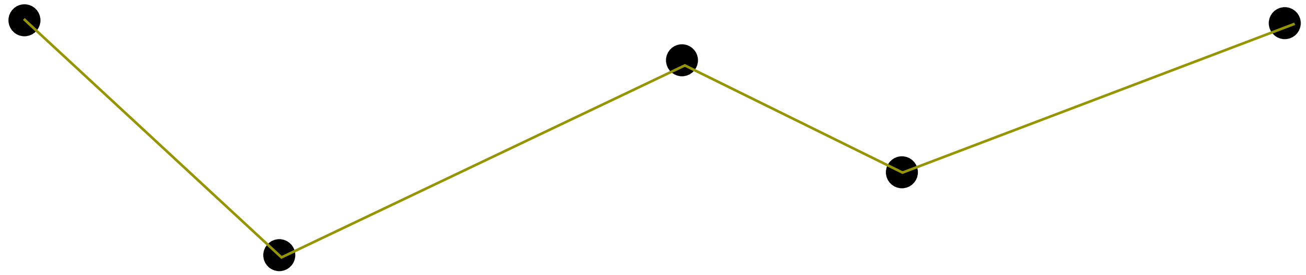
Douglas–Peucker Algorithm



Douglas–Peucker Algorithm



Douglas–Peucker Algorithm



Running Time

```
function DouglasPeucker(P[], ε)
  // Find the point with the maximum distance (=maximum cross product)
  cmax = 0
  index = 0
  for i = 2 to ( end - 1)
    c = P[1]P[n] × P[1]P[i]    (vectors)
    if ( c > cmax )
      index = i
      cmax = c
  if (cmax / ||P[1][P[n]|| > ε) {
    R1 = DouglasPeucker(P[1..index], ε)
    R2 = DouglasPeucker(P[index..n], ε)
    R1.removeLast
    return R1 || R2 // Concatenate the two lists
  else
    return [P[1], P[n]] // Only return the first and last points
```

$c = \overrightarrow{P[1]P[n]} \times \overrightarrow{P[1]P[i]}$

$\left\| \overrightarrow{P[1][P[n]} \right\|$

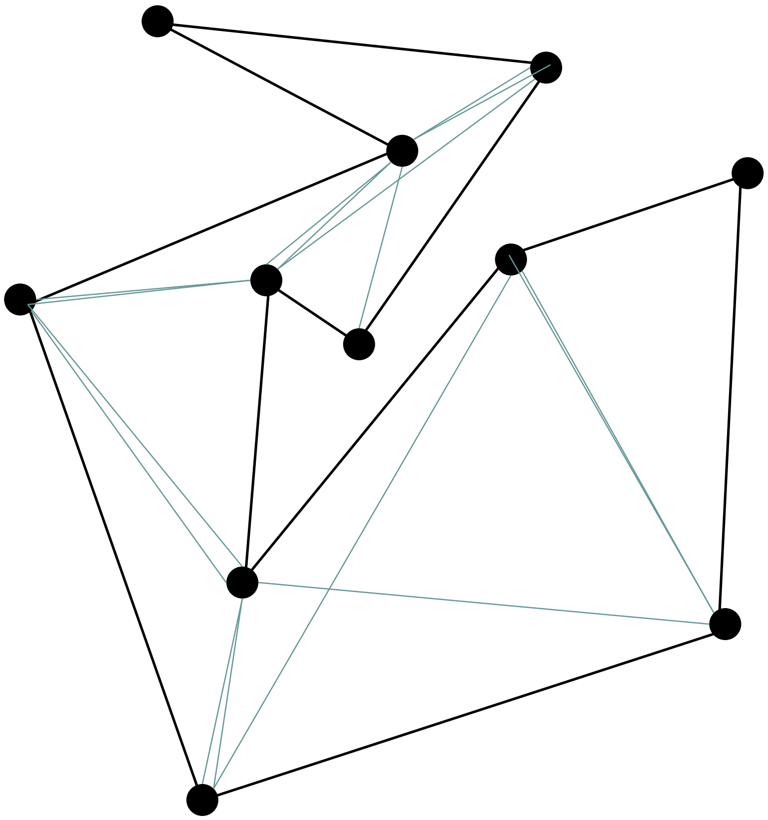
$$T(n) = T(n1) + T(n - n1) + O(n)$$

Polygon Triangulation

Polygon Triangulation

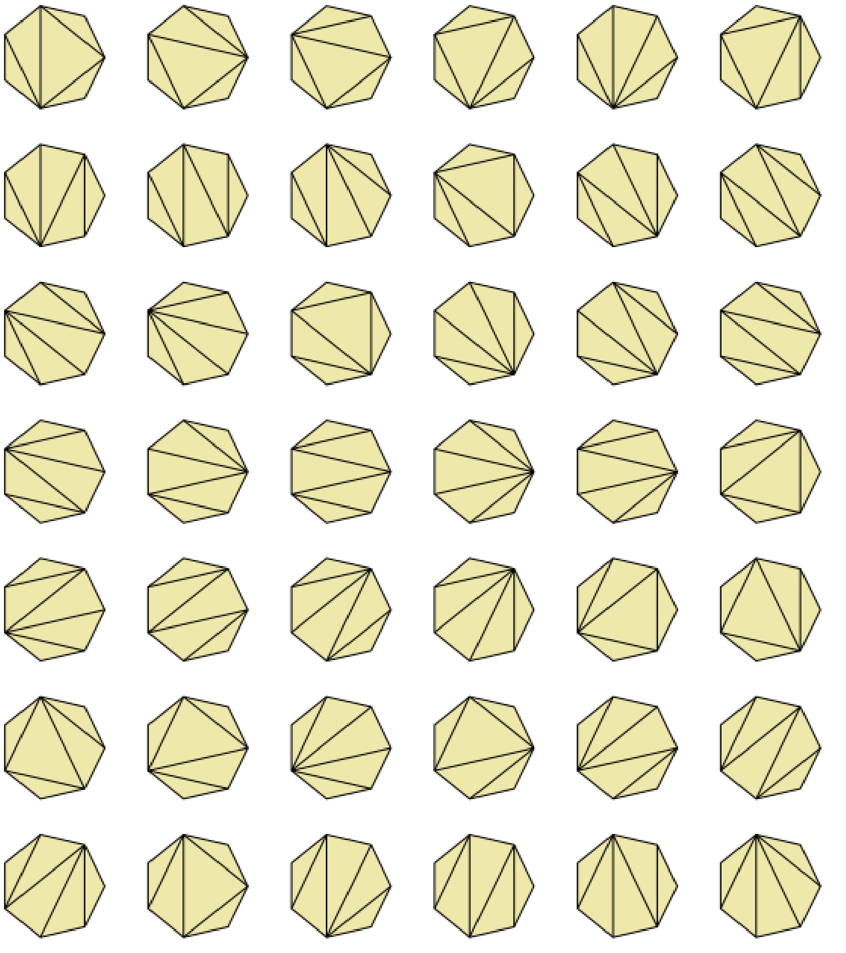
- Given a simple polygon P, break it down into a set of triangles T such that the union of the triangles is equal to the polygon and no two triangles intersect. That is: $\bigcup_{t_i \in T} t_i = P$ and $t_i \cap t_j = \phi$ for any $t_i, t_j \in T$ and $i \neq j$

Polygon Triangulation



Convex Polygons

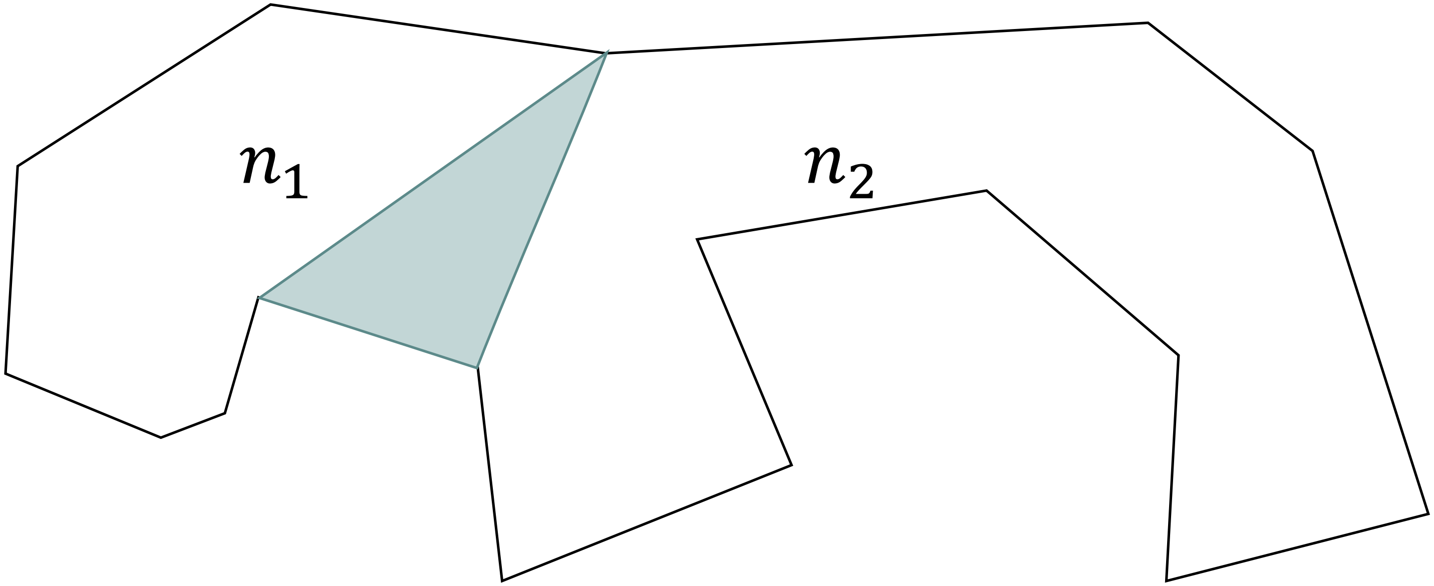
- Choose any vertex on the polygon
- Connect it to all other vertices to create all diagonals
- Number of possible triangulations is Catalan Number C_{n-2}
- $C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{2n!}{(n+1)!n!}$



Number of Triangles

- Number of triangles in a triangulation of a polygon of n points is $n - 2$ triangles
- Can be proven by induction
- Trivial case: $n = 3$
- General case: If it applies for all $n < m$, we need to prove that it applies for $n = m$
- That is, we want to prove that if a triangulation exists, it will have $m - 2$ triangles

Induction

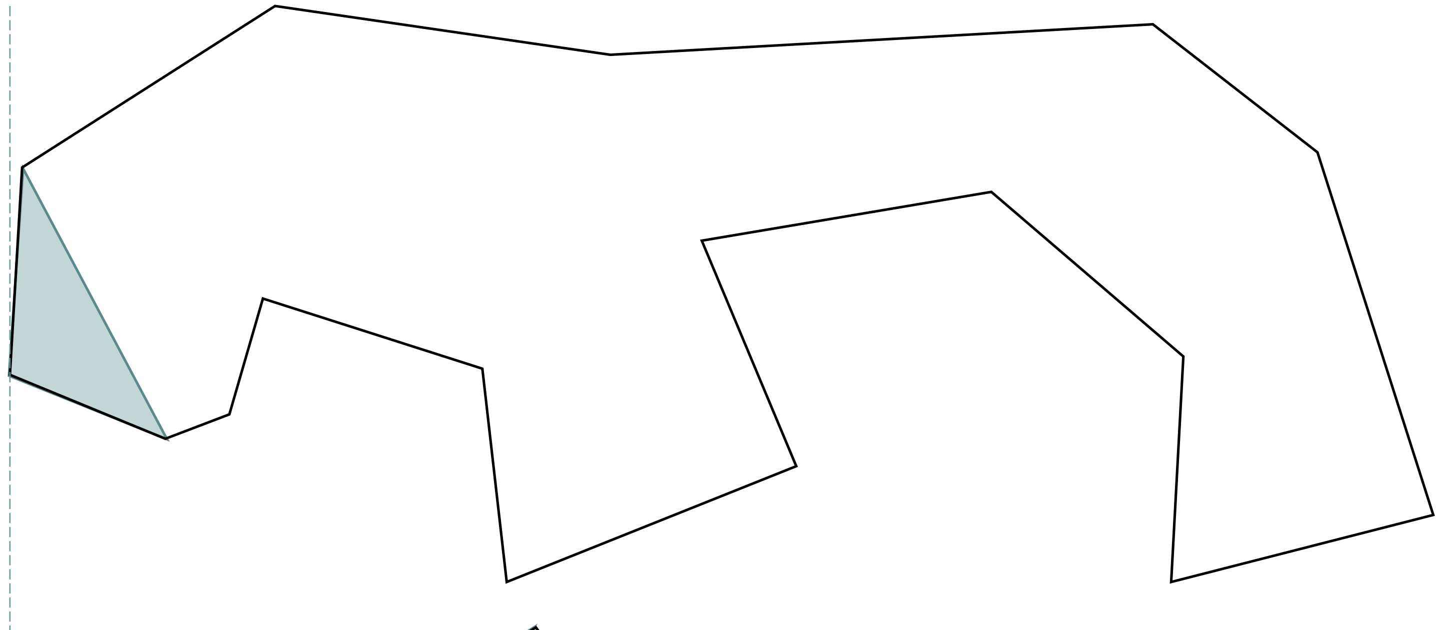


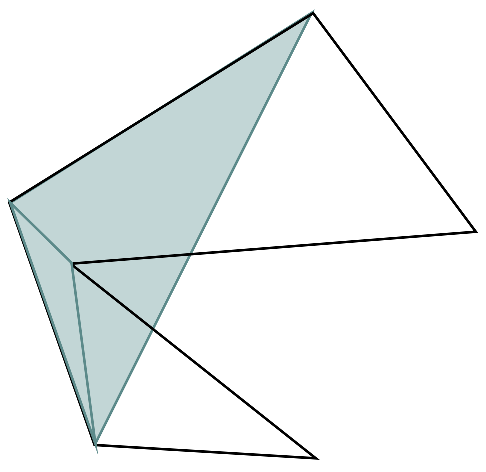
- $n_1 + n_2 = m + 1$
- # of triangles in $P_1 = n_1 - 2$, in $P_2 = n_2 - 2$
- # of triangles in $P = n_1 - 2 + n_2 - 2 + 1$
- $= n_1 + n_2 - 4 + 1 = m + 1 - 4 + 1 = m - 2$

Existence of a Triangulation

- Any simple polygon has at least one triangulation
- Proof by induction
- Trivial case: $n = 3$
- General case: If there are triangulations for all polygons $n < m$, we need to prove that there is one for $n = m$

Induction





Dual Graph

- Dual graph $\mathcal{G}$
- Each triangle is represented by a vertex in $\mathcal{G}$
- Two triangles that share an edge are connected by an edge in $\mathcal{G}$