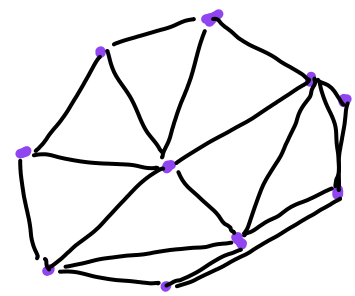
13 January 2025

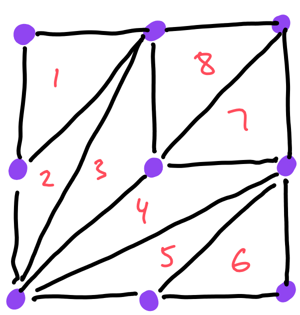
TRIANGULATIONS OF POINTS (IN THE PLANE)

Def: A **TRIANGULATION** of a planar point set S is a subdivision of the plane determined by a maximal set of non-crossing edges whose vertex set is S.

you can't add more



Example: Let S consist of 9 points in a 3×3 grid

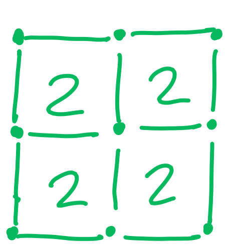


How many triangles were formed? 8

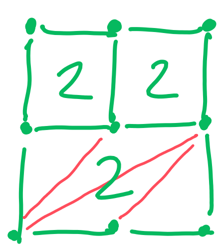
Is it always 8?

How many triangulations are possible for this point set?

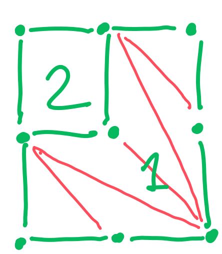
Count triangulations by which horizontal/vertical edges are present:



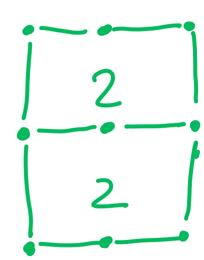
$2^4 = 16$ triangulations



8 triangulations
$\times 4$ rotations
32



2 triangulations
$\times 4$ rotations
8



4 triangulations
$\times 2$ rotations
8

$\boxed{\text{Total triangulations: } 64}$

BONUS: 4x4 grid: 18 triangles;
46,456 triangulations

5x5 grid: 32 triangles;
736,983,568 triangulations

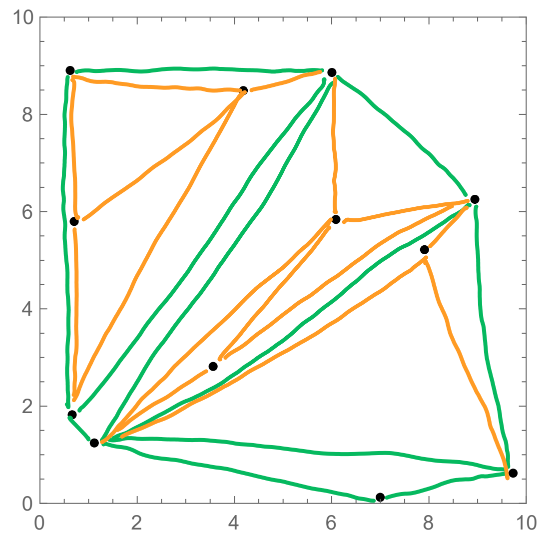
Triangulation Algorithms

MATH 261 Computational Geometry

Consider the following algorithm for triangulating a set of points S in the plane.

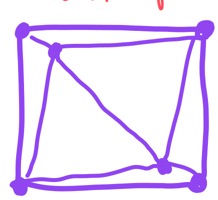
> Triangle-Splitting Algorithm: Find the convex hull of S and triangulate this hull as a polygon. Choose an interior point and draw edges to the three vertices of the triangle that contains it. Continue this process until all interior points are exhausted.

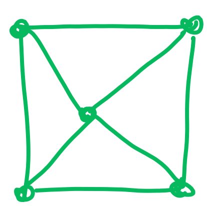
1. Use the triangle-splitting algorithm to triangulate the following point set S:



2. Depending on how you triangulate the convex hull and the order in which you choose the interior points, can the triangle-splitting algorithm produce *all* possible triangulations of S?

NO: counterexamples:





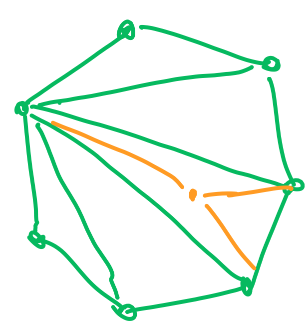
3. What is the computational complexity of the triangle-splitting algorithm?

find hull: $O(n \log n)$

triangulate the hull: $O(n)$

locate interior points: $O(n^2)$

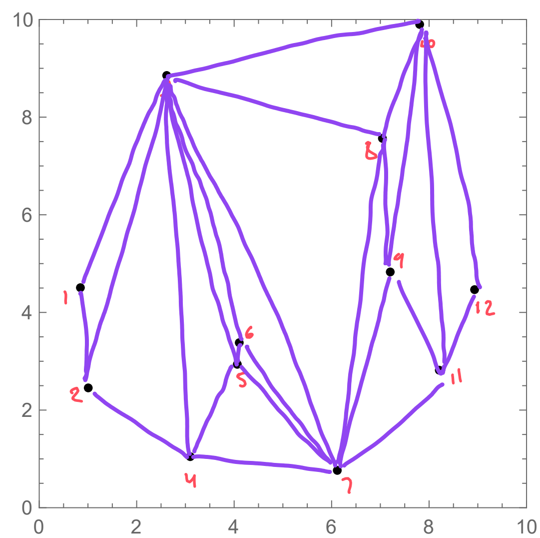
$\to O(n \cdot \log(n))$ possible with more work/data structures



Overall: $O(n \cdot \log(n))$

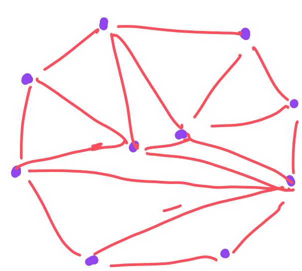
Incremental Algorithm: Sort the points of S by their x-coordinates. The first three points determine a triangle. Consider the next point p in the ordered set and connect it with all previously considered points $\{p_1, \ldots, p_k\}$ which are visible to p. Continue this process of adding one point of S at a time until all of S has been processed.

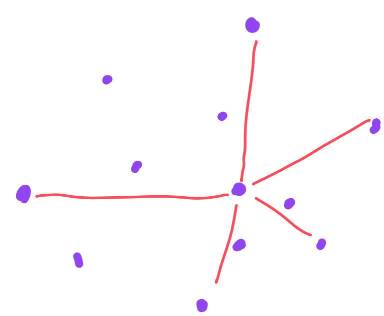
4. Use the incremental algorithm to triangulate the following point set S:



5. If you are allowed to rotate the set of points arbitrarily before applying the incremental algorithm, does the algorithm produce *all* possible triangulations of S?

NO





6. What is the computational complexity of the incremental algorithm?

Sort points: $O(n \cdot \log(n))$

Add n points, for each find the visible vertices in $O(n)$ time: $\Big\}\ O(n^2)$

Overall: $O(n^2)$