Definitions

- A **polygonal chain** is a finite concatenation of line segments where consecutive segments share one end point ("One ends where the next starts").
- A **simple polygon** is a "closed polygonal chain that doesn't self-intersect".
  - Simple polygons have **vertices** and **edges** that form their **boundaries**.
  - A **diagonal** is a line segment that connects two vertices.
- A **triangulation** of a simple polygon is a decomposition of it into triangles by cutting along a maximal set of diagonals.
  - There could be many different triangulations.

An example of a triangulation
Theorem 3.1

1. Every $n$-sided simple polygon has (at least) one triangulation.
2. Each triangulation has $n-2$ triangles.

Proof:
- Induction over the size $n$.
- There always exists a diagonal.
  - $\Rightarrow$ It’s possible to triangulate.
  - A counting argument.

About triangulations in 2D

- One could also triangulate a set of points.
- In the plane (2D), there always exists a triangulation of a simple polygon or point set.

What about 3D?

- In 3D, where triangles become pyramids (tetrahedras), there exist polyhedra that cannot be partitioned (tetrahedralized) into pyramids.
  - It is NP-complete to determine whether or not a given polyhedra can be decomposed into pyramids.

Guarding an Art Gallery

Figure 2: An unpartitionable polyhedron.
Guarding an Art Gallery

- Triangulate the (polygonal drawing of the) art gallery.
- One guard (or surveillance camera) per triangle is sufficient to guard a triangulated simple polygon.
  - Two points in a polygon are visible to each other if the line segment that connects them lies in the polygon.
  - Each camera covers one triangle.
  - Takes $n/2$ guards.
- Could we do with less guards?

Guarding an Art Gallery (cont.)

- 3-color the vertices of a triangulation!
  - Each triangle has a black, grey, and white vertex.
- Assume that there are at least black vertices (there are roughly one third of each).
- Then place guards at black vertices.
  - $\Rightarrow \lceil n/3 \rceil$ guards!
- Is there always a 3-coloring?
  - YES. Look at the dual graph of the triangulation.
  - It is a tree with nodes for triangles and arcs between nodes whose triangles share sides (are joined along a diagonal).
- Could we do even better?
  - NO. There are polygons that require $\lceil n/3 \rceil$ guards.
**The Art Gallery Theorem**

- $\lceil n/3 \rceil$ guards are necessary and sufficient to guard an $n$-sided simple polygon.

**Computing a triangulation - Version 1**

- Find the leftmost vertex $v$.
- a) If a diagonal exists that connects the neighboring vertices to $v$,
  - Cut out the triangle containing $v$ along the diagonal.
  - Triangulate the remaining polygon.
- b) else
  - Find a diagonal that connects $v$ with another vertex $w$:
    - Such a diagonal can be found in $O(n)$ time.
    - Split the polygon in two along the diagonal.
    - Triangulate the parts (recur).
- $O(n^2)$ time.

**Computing a triangulation - Version 2**

- Overall description:
  - Partition the polygon into monotone pieces in $O(n \log n)$ time.
    - We will use plane sweep where the vertices are event points.
  - Triangulate each piece separately in a total of $O(n)$ time.
    - The algorithm is based in the greedy technique.
    - Put together all triangulations into one in $O(n)$ time.
- A simple polygon is monotone, with respect to a line if the intersection between the polygon and any line perpendicular to the line is connected.
  - We will compute $y$-monotone pieces.
- Doubly-Connected Edge Lists

**Definitions**

- $\square$ = start vertex
- $\blacksquare$ = end vertex
- $\bigcirc$ = regular vertex
- $\triangle$ = split vertex
- $\triangledown$ = merge vertex
During the (vertical) plane sweep

- Split vertices:
  - Split vertices:
  - Merge vertices:

Triangulating a monotone polygon

- Merge all vertices by their $y$-coordinate into $u_1, u_2, u_3, \ldots, u_n$ (leftmost first, if ties).
- Push $u_1$ and $u_2$ onto an empty stack $S$.
- FOR $j := 3$ TO $n-1$ DO
  - IF $u_j$ and the top of $S$ lie on different sides of the polygon then Case A
  - else Case B.

Case A

- Pop all vertices from $S$.
- Report a diagonal between $u_j$ and each of the popped vertices except the last one popped.
- Push $u_j$, and $u_j$ onto $S$.

Case B

- Pop one vertex from $S$.
- Pop vertices from $S$ as long as the diagonal between the vertices and $u_j$ lie inside the polygon.
  - Report a diagonal per vertex.
- Push the last vertex back onto $S$.
- Push $u_j$ onto $S$. 
Triangulating general subdivisions

- Partition into monotone pieces and triangulate as in the case of simple polygons.

- Theorem 3.9: A planar subdivision with $n$ vertices in total can be triangulated in $O(n \log n)$ time using $O(n)$ space.