Last Time: PSLGs

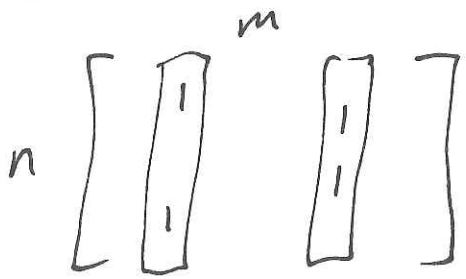
Topology of embedded planar graphs:
- Jordan Curve Thm (for Polygons)
- Euler's Formula

Today: Data Structures for PSLGs

Goal: store a PSLG that supports "walking around", i.e. incidence operations.

Warmup: Let's start in 1D ⇒ Incidence operations
- (i) edges incident to a vertex.
- (ii) vertices incident to an edge

Idea 1: Edge-Vertex Incidence Matrix



Size: $O(nm)$

Operations: $O(n)$ & $O(m)$

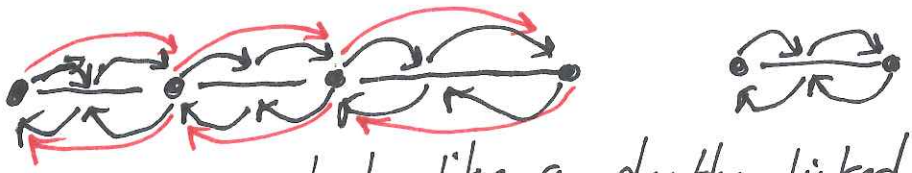
Big + Slow

Advantage: Works to store more general graphs.

However: Euler's Formula ⇒ planar graphs are <u>sparse</u> (avg degree is constant)

Idea 2: Just store neighbors (nbrs).

- ≤2 edges per vertex
- ≤2 vertices per edge

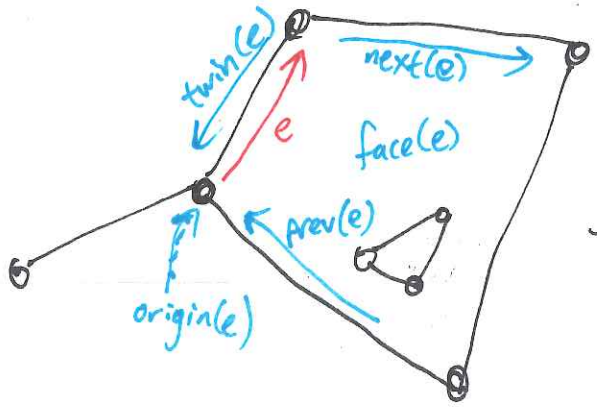


Starts to look like a doubly linked list.

Add a bit of geometry: Distinguish left and right.

Let's do this in the plane.

Doubly-Connected Edge List (A.K.A. Half-edge Data structure)



Convention: face(e) is on the right.

One Edge ⇝ Two Half-edges (one in each direction)

Store V, E_{half}, F

$v \in V$: $e \in E$ s.t. $v = origin(e)$

$f \in F$: $e \in E$ s.t. $f = face(e)$ (one for each bdy component)

$e \in E$: origin, face, twin, next, previous

Incidence Operations:

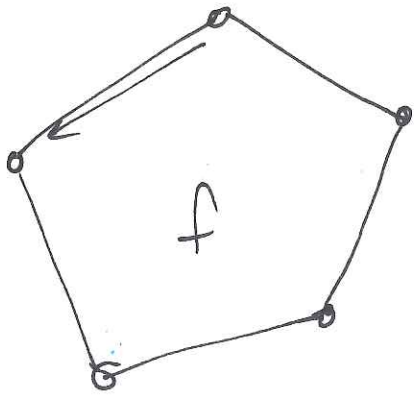
I. List edges incident to a vertex v (in ccw order)

```
Let  e = halfedge(v).
Let  e' = e
Let  output = ∅
Do
     output << e'
     let  e' = twin(prev(e'))
Loop while (e' ≠ e)
return output.
```

have to start somewhere!

II List edges in bdy of a face

First consider a face with **No Holes.**




```
Let  e = he(f)
Let  e' = e
Let  output = ∅
Do
   output << e'
   let  e' = next(e')
Loop while (e' ≠ e)
return output
```

If there are holes, repeat for each hole.

Restricting Ourselves to Graphs with some minimum Connectivity.

Def A graph G is **k-connected** (a.k.a. k-vertex connected) if G has $> k$ vertices and $G - X$ (induced subgraph) is connected for every set X of $< k$ vertices.

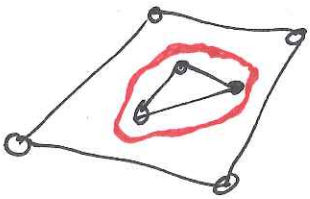
---

ex) 1-connected = connected (i.e. there is a path between every pair)

2-connected also called biconnected

k-connected $\Rightarrow (k-1)$-connected

For connected graphs, we only need to store one halfedge for each face because no faces have "holes."



For ~~connected~~ 2-connected graphs, All bdys are simple polygons