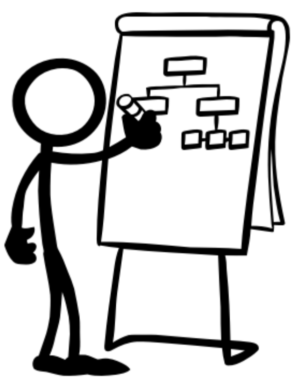
236719 Computational Geometry – Tutorial 1

Planar Graph
Euler's Formula
DCEL

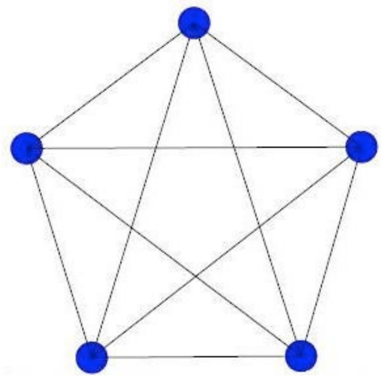


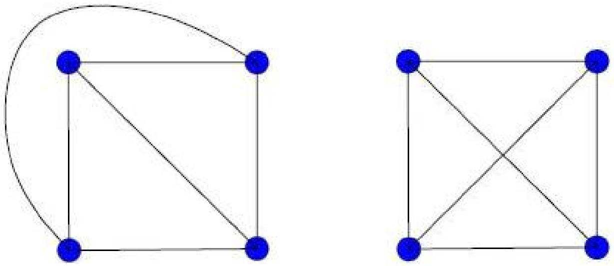
Amani Shhadi

Based on slides by
Yufei Zheng - 郑羽霏

Planar Graph

- **Definition** – A **planar graph** is a **graph** that can be embedded in the plane
  - Can be drawn on a plane in such a way that its edges intersect only at their endpoints
  - In some pictures, a planar graph may have crossing edges





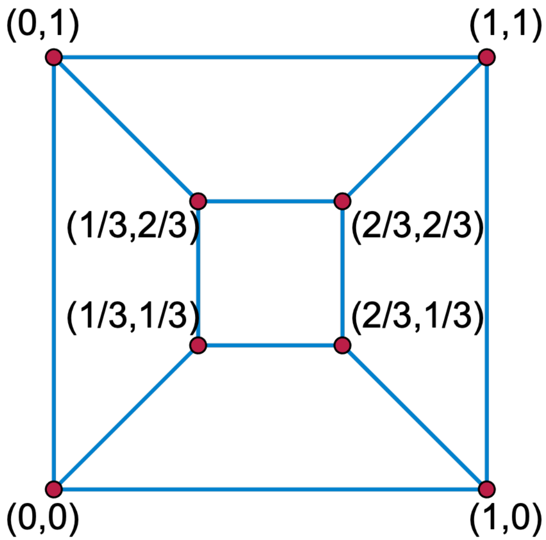
Simple graph:

- undirected
- no graph loops
- no parallel edges

Planar Graph

- **Fáry's Theorem** - every *simple planar graph* admits an embedding in the plane such that all edges are straight line segments which don't intersect.

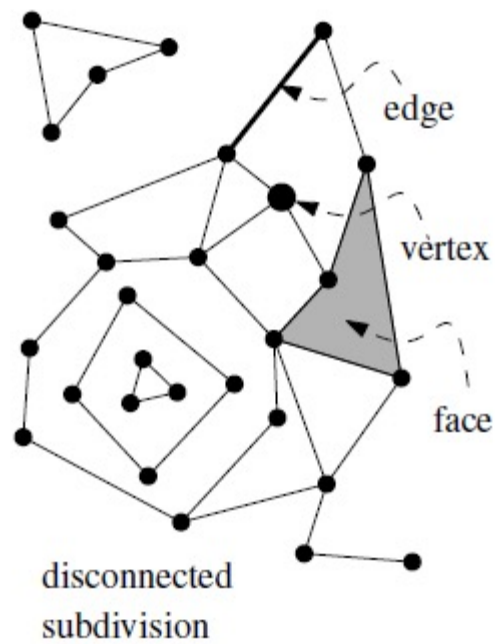
- **Tutte Embedding** - the embedding of 3-vertex-connected planar graphs with good properties.



Euler's Formula

Notations

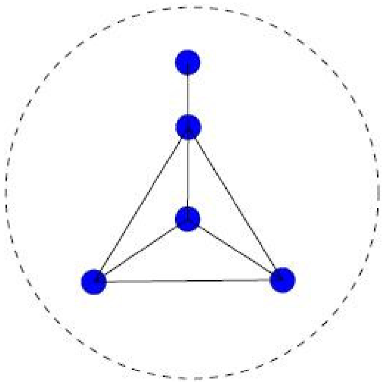
- v – number of vertices
- e – number of edges
- f – number of faces



Euler's Formula (for finite, connected planar graph)

$$v - e + f = 2$$

$v =$
$e =$
$f =$



$v - e + f = 2$ – Proof by induction on the complexity of the graph

- Base case: $f = 1$

  acyclic connected graph – Tree

  $$e = v - 1$$

  $$v - e + f = v - (v - 1) + 1 = 2$$

$v - e + f = 2$ – Proof by induction on the complexity of the graph

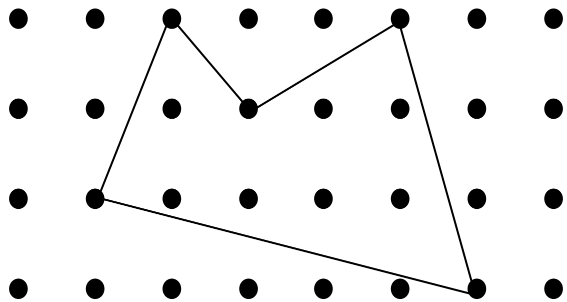
◎Induction step: Consider a graph with f' faces, v' vertices and e' edges.
Assume that the property holds for $f = f' - 1$

a. Choose an edge that is shared by 2 different faces and remove it, the graph remains connected.
b. This removal decreases both the number of faces and edges by one, on the new graph we get:

$$v - e + f = 2$$
$$v - e + f = v' - (e' - 1) + (f' - 1) = 2$$
$$\Rightarrow v' - e' + f' = 2$$

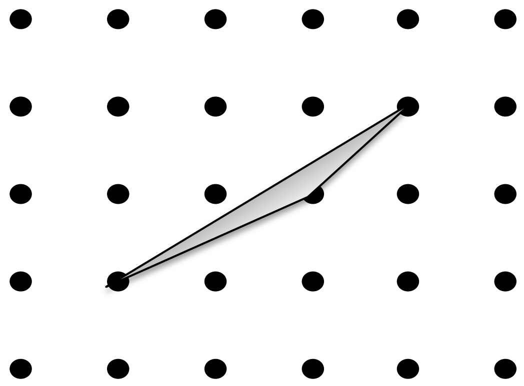
Applications of Euler's Formula – Pick's Theorem

- What is the area of this polygon?



Applications of Euler's Formula – Pick's Theorem

- Let us begin with a simpler case, what is the area of a triangle containing no inner points:



- Lemma: the area of a triangle containing no inner points is $\frac{1}{2}$.

Applications of Euler's Formula – Pick's Theorem

- A basis of $\mathbb{Z}^2$ is a pair of vectors e_1, e_2 such that
$$\mathbb{Z}^2 = \{\lambda_1 e_1 + \lambda_2 e_2 \mid \lambda_1, \lambda_2 \in \mathbb{Z}\}$$

- **Lemma:** If $\{(x_1, y_1), (x_2, y_2)\}$ is a basis of $\mathbb{Z}^2$ then $\det(A) = \pm 1$ where $A = \begin{pmatrix} x_1 & x_2 \\ y_1 & y_2 \end{pmatrix}$
- **Proof:**

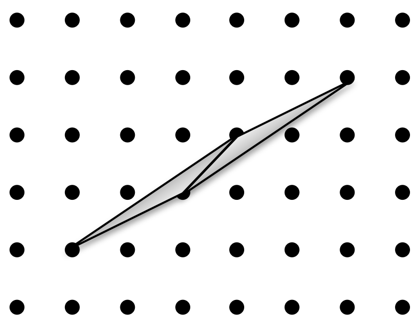
  There exists a matrix Q s.t. $AQ = I$

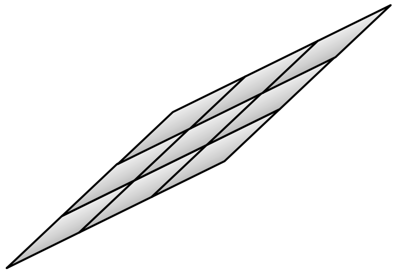
  $\Rightarrow \det(A) \cdot \det(Q) = 1$

  All the numbers are integers, hence the result.

Applications of Euler's Formula – Pick's Theorem

- **Lemma:** If the triangle created by a pair of vectors contains no lattice points, this pair is a basis of $\mathbb{Z}^2$.





- **Corollary:** The area of a lattice triangle containing no inner points is $\frac{1}{2}$.

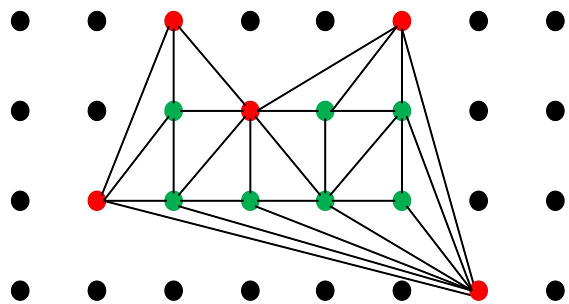
Applications of Euler's Formula – Pick's Theorem

- **Pick's theorem:** The area of a polygon Q, with integral vertices is given by

$$A(Q) = n_{int} + \frac{1}{2} n_{bd} - 1$$

Where:

- n_{int} is the number of interior points
- n_{bd} is the number of boundary points.



$n_{int} = 7$
$n_{bd} = 5$
$A = 8.5$

Applications of Euler's Formula – Pick's Theorem

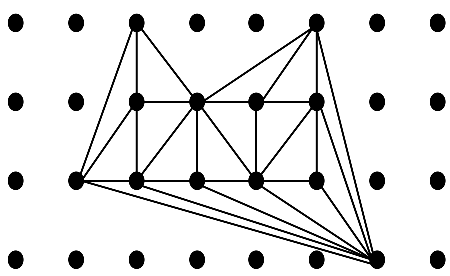
- Number of triangles: $f-1$
- Number of boundary edges: e_{bd}
- Number of interior edges: e_{int}

$$3(f-1) = 2e_{int} + e_{bd}$$
$$\Rightarrow f = 2(e-f) - e_{bd} + 3$$
$$= 2(n-2) - n_{bd} + 3$$

Euler's Formula

$$= 2n_{int} + n_{bd} - 1$$

$n = n_{int} + n_{bd}$



$$A(Q) = \frac{1}{2}(f-1)$$

$v - e + f = 2$

Applications of Euler's Formula

- Exercises:

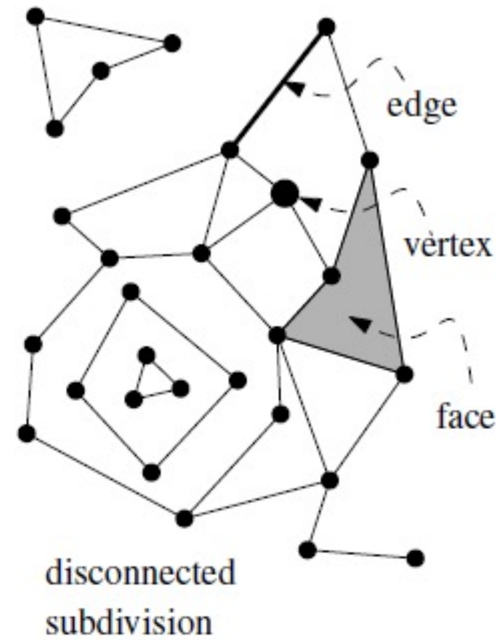
Show that for any planar graph:

- Have at most $3V - 6$ edges.
- Have a vertex of degree at most 5.

DCEL – Doubly Connected Edge List

- Given a planar graph we are looking for a DS to represent the graph.

- We want to enable (for example):
  - Traverse all edges incident to a vertex v
  - Traverse all edges bounding a face
  - Traverse all faces adjacent to a given face
  - etc…



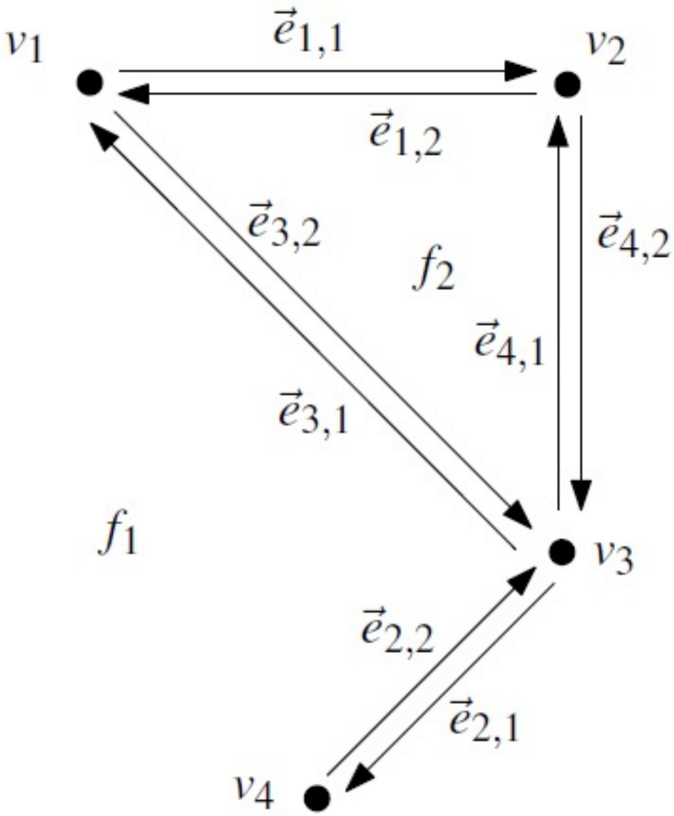
DCEL – Doubly Connected Edge List

- Complexity of a subdivision $= V + E + F$
- DCEL – A data structure for representing an embedding of a planar graph in the plane
  - Only consider: every edge is a straight line segment
  - Recall Fáry's Theorem

- DCEL consists of 3 collections of records: Vertices, Edges, Faces

DCEL – A Record for Vertex

- **Vertex** – the embedding of a node of the graph
- **Coordinates(v)** – coordinates of vertices
- **IncidentEdge(v)**
  - Points to only one edge

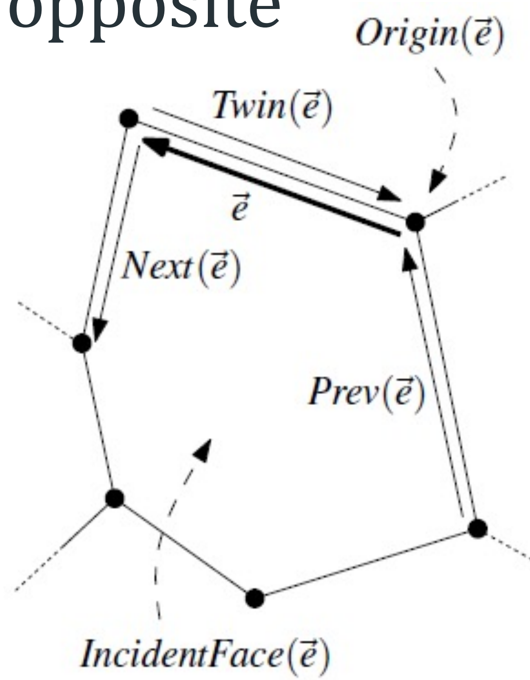


Vertex	Coordinates	IncidentEdge
v_1	$(0,4)$	$\vec{e}_{1,1}$
v_2	$(2,4)$	$\vec{e}_{4,2}$
v_3	$(2,2)$	$\vec{e}_{2,1}$
v_4	$(1,1)$	$\vec{e}_{2,2}$

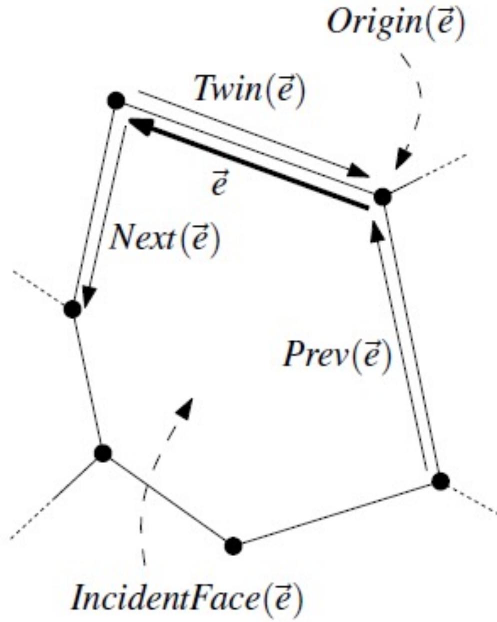
DCEL – A Record for Edge

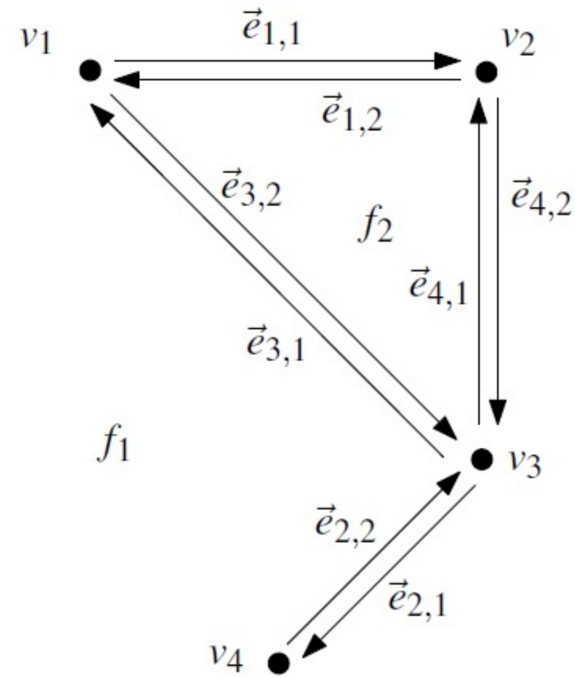
- **Half-edges** – different sides of an edge
  - Bounds only 1 face
- **Origin(e)**
  - **Orientation** – the face it bounds lies to its left
  - ID of a vertex structure
- **Twin(e)** – the twin edge of e in the opposite direction
- **IncidentFace(e)**
- **Next(e)** & **Prev(e)**

  next and previous edge on the boundary of *IncidentFace*(*e*).



DCEL – A Record for Edge



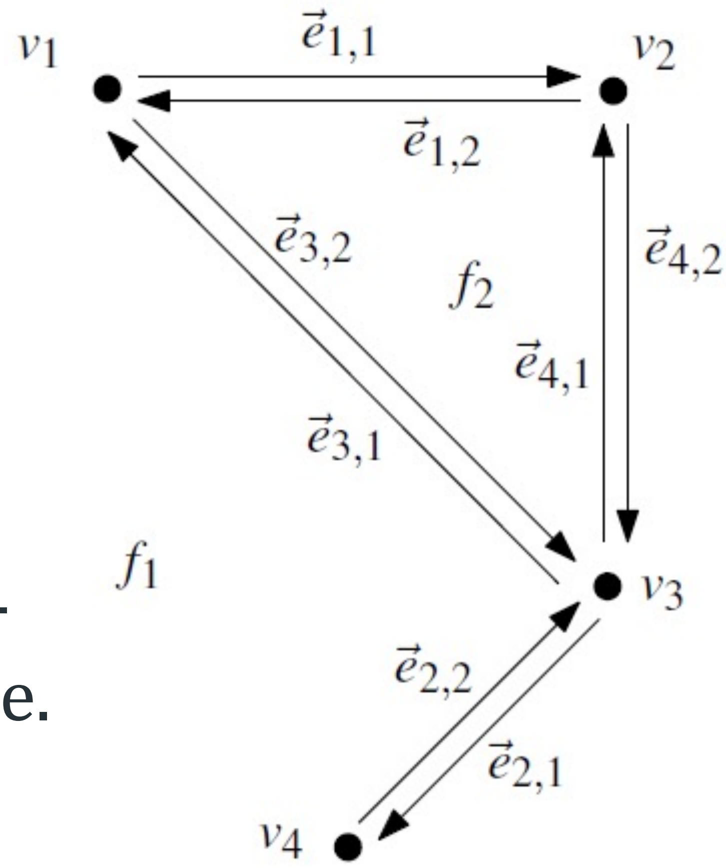


Half-edge	Origin	Twin	IncidentFace	Next	Prev
$\vec{e}_{1,1}$	v_1	$\vec{e}_{1,2}$	f_1	$\vec{e}_{4,2}$	$\vec{e}_{3,1}$
$\vec{e}_{1,2}$	v_2	$\vec{e}_{1,1}$	f_2	$\vec{e}_{3,2}$	$\vec{e}_{4,1}$
$\vec{e}_{2,1}$	v_3	$\vec{e}_{2,2}$	f_1	$\vec{e}_{2,2}$	$\vec{e}_{4,2}$
$\vec{e}_{2,2}$	v_4	$\vec{e}_{2,1}$	f_1	$\vec{e}_{3,1}$	$\vec{e}_{2,1}$
$\vec{e}_{3,1}$	v_3	$\vec{e}_{3,2}$	f_1	$\vec{e}_{1,1}$	$\vec{e}_{2,2}$
$\vec{e}_{3,2}$	v_1	$\vec{e}_{3,1}$	f_2	$\vec{e}_{4,1}$	$\vec{e}_{1,2}$
$\vec{e}_{4,1}$	v_3	$\vec{e}_{4,2}$	f_2	$\vec{e}_{1,2}$	$\vec{e}_{3,2}$
$\vec{e}_{4,2}$	v_2	$\vec{e}_{4,1}$	f_1	$\vec{e}_{2,1}$	$\vec{e}_{1,1}$

DCEL – A Record for Face

- **OuterComponent(f)** – A **pointer** to a half-edge on the outer boundary of face f.

- **InnerComponents(f)** – A **list** contains for each hole in the face f a pointer to some half-edge on the boundary of the hole.



Face	OuterComponent	InnerComponents
f_1	nil	$\vec{e}_{1,1}$
f_2	$\vec{e}_{4,1}$	nil

DCEL – Further Facts

- **Amout of Storage** – linear in the complexity of the subdivision
  - vertices and edges – linear in $V + E$
  - faces
    - OuterComponent – linear in F
    - InnerComponent lists– linear in E

- **Special cases**
  - For Isolated vertices in a face, store pointers
  - For additional information, add attributes

DCEL – Exercises

- Why isn't the *Destination* field of the |
  Edge structure needed?

  Origin(Twin(e))

- Traverse all edges incident to a vertex v

```
e1 = IncidentEdge(v)
do:
e1 = Next(Twin(e1))
While e1 != IncidentEdge(v)
```