What do we think about graph?

Discrete math: Abstract vertices + edges being pair of vertices

Topology: 1-dimensional geometric object

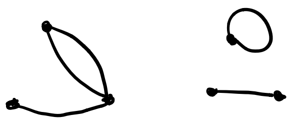
What else do we care about graph in computational topology?

There is a one-to-one correspondence between hypereelliptic surfaces of a given genus and a set of graph.

finite in our case

Def A (simple) graph is a pair $G=(V,E)$ where V is a set whose elements are called vertices, and $E \subseteq \binom{V}{2}$.

I.e. $E = \{\{a,b\} \mid a,b \in V,\ a \neq b\}$.



not simple graph.

Def ① let $u,v \in V$. a <u>path</u> between u and v is a seq of vertices $u=u_0, u_1, \ldots, u_{n-1}, u_n = v$ s.t. $\{u_i, u_{i+1}\} \in E$ for $i=0,\ldots,n-1$.

The path is <u>simple</u> if $u_0, \ldots, u_n$ are pairwise distinct.

② A graph is connected if there is a path between every two vertices.

explain sth for subgraph

③ A connected component is a maximal subgraph that be connected.

Def A <u>tree</u> is a graph with a unique simple path between every pair of vertices.

In particular, a tree is connected. Deleting any edge of a tree disconnects the tree.

Prop Let $G=(V,E)$ be a connected graph, then TFAE:

① G is a tree.

② $\operatorname{card} E = \operatorname{card} V - 1$.

④ G has no <u>circuit</u>, i.e. a path $u_0, u_1, \dots, u_n$ with $n \geq 1$ and $u_i = u_j$ only when $\{i,j\} = \{0,n\}$.

Pf: ① ⇒ ② By induction, deleting one edge and got 2 trees. Then add that edge. (Clearly, a connected subgraph of a tree is a tree.)

② ⇒ ③ If there is a circuit $u_0, u_1, \dots, u_n$, $u_0 = u_n$, then the graph has at least $\operatorname{card} V$ edges (add one additional point need at least one edge to connect the graph)

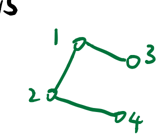
③ ⇒ ① If there are two simple paths, there is definitely a circuit.

Def Let $G=(V,E)$ be a graph. A <u>spanning tree</u> $T=(V,F)$ of G is a tree with $T \subseteq E$.

G is connected if and only if it has a spanning tree.

Disjoint Set Systems

Problem: Given a graph $G=(V,E)$, we want to find its connected components.



→ $V = [\![1,4]\!]$, $E = \{\{1,2\},\{2,4\},\{1,3\}\}$

Easy to visualize when V, E are small. But what if the graph is complicated? How do a computer know?

Set up: Data structure of V: list of vertices $[v_1, \dots, v_n]$ (math sage, python, Arraylist in Java). We'll use $v_i = i$ for convenient.

E: list of tuples $[(v_i, v_j), \dots]$

Kruskal's algorithm:

Input: $G=(V, E)$.

Output: Spanning trees of components of G. /

Disjoint subsets of V.

Step 0: Create singeleton sets for all $i \in V=[1, 2, \dots, n]$.

All we do is to convert an element to a set.

Step 1: Consider $E[1]=(i, j)$ and union $\{i\}, \{j\}$ into $\{i, j\}$.

Label $\{i, j\} = i$ or j, whatever. Now $la(i) = la(j) = i$.

Label of i is the name of the set where i live in.

⋮

Step k: Consider $E[k]=(i_k, j_k)$, find $la(i_k)$, $la(j_k)$.

If $la(i_k)=la(j_k)$, do nothing (we are discarding this edge)

Else, union $la(i_k)$ and $la(j_k)$ That's why we get spanning tree!

and set new common lables for all elements in the union

After card E steps, we get disjoint subsets of V.

Q: ① How do we store the label information in an integer data?

In other words, how to determine in which set i lives in?

Have no choice but to search every set one by one!

② How to choose a representative? – Randomly?

③ Where is the spanning tree structure !?

– How to store a tree in computer

Define a new data type: Node, representing vertices

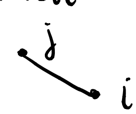
Better to be a pointer.

In C, C++: struct Node { int value; Node parent; }

In C++, Java, Python: class Node { int value; Node parent; }

Step 0: For $i \in V$, set new Node i with value i, parent null

Step 1: Consider $(i, j) \in E$, put Node i. parent $= j$



⋮

Step k: Consider $(i_k, j_k) \in E$. Let $x = \text{find}(i_k)$, $y = \text{find}(j_k)$.

where the find function:

find (Node i) { (Input: one node)

If Node i . parent = null, return i . endif

Else return find (Node i . parent)

(Output: the root of the tree where i lives in)

If $x = y$, do nothing (discarding the redundant edges)

If $x \neq y$, union (x, y)

where the union function:

Union (Node i, Node j) { (Input: Two (root) nodes)

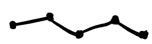
If i. size > j. size, $i \leftrightarrow j$ endif

Set i. parent = j.

Process end after card E steps.

Result: Get connected components and some of the edges in each components (by parenting relationships)

 So got spanning trees.

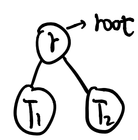
Improving the running time: Currently, to find the root of a node in a tree of size k, we need to traverse k−1 steps in the worst case. () The recursive algorithm is fairly slow.

By implementing the purple part, we limit this number to $\log_2 k$.

Because: By induction on the size of the tree.

At the very first step, when $k = 1, 2$, this is true.

Each time we join a smaller tree to a bigger tree:



$|T_1| \leq |T_2|$. (Originally T_1 & r — T_2) size $|T_1| + |T_2| + 1$

$\forall v \in T_2$, by induction hypothesis, it takes at most $\log_2 |T_2|$ steps to reach the root of T_2, namely r. $\log_2 |T_2| \leq \log_2 (|T_1| + |T_2| + 1)$

So things is true for nodes in T_2;

$\forall u \in T_1$. by induction hypothesis it takes at most $\log_2 |T_1|+1$ steps to reach r. But $\log_2 |T_1|+1 = \log_2(2|T_1|) \le \log_2(|T_1|+|T_2|+1)$.

So we are done.

By doing some parallelization. the running time could be limited under $m\alpha(n)$. where m is a constant and $\alpha(n)$ is the inverse of the Ackermann function.

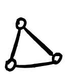
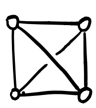
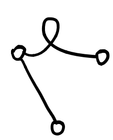
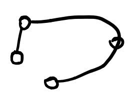
($\alpha(n) = A^{-1}(n)$. where $A(4) = 2^{2^{2^{65536}}} - 3$) So α grows extremely slow and can be regarded as a constant in practice.

Planar graphs

Def ① Let $G=(V,E)$ be a graph. A <u>drawing</u> maps every vertex $v \in V$ to a point $f(v) \in \mathbb{R}^2$. and every edge $(u,v) \in E$ to a path in $\mathbb{R}^2$ with endpoints $f(v)$ and $f(u)$.

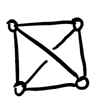
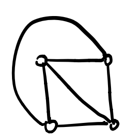
② The drawing is an <u>embedding</u> if $\{f(v) \mid v \in V\}$ are pairwisely distinct and the paths satisfying:

(i) Simple (ii) Have no intersection with other paths nor other vertex in their interior.

E.g.  embedding;  .  ,  not embedding

③ The embedding is called a <u>straight line embedding</u> if every path is a straight line.

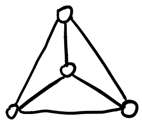
Def A graph is a <u>planar graph</u> if it has an embedding.

E.g. The underlying graph of  is planar since  is an embedding.

One goal for this section: Show that every planar graph has a straight-line embedding.

More about planarity

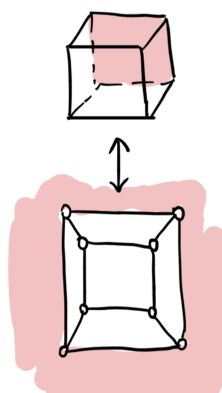
Def A <u>face</u> for an embedding is a component in the decomposition of space defined by the embedding.

E.g.  has 4 faces. 3 of which bounded.

Thm (Euler's relation for planar graphs) Every embedding of a conneited planar graph $G=(V,E)$ satisfies $n-m+l=2$. where $n = \text{card } V$, $m = \text{card } E$ and $l = \#$ of faces.

Comments: ① This implies that # of face is an invariant for different embeddings of a planar graph.

② Recall the Euler characteristic of a polyhedron: $n-m+l=2$. Vertex, edge, face, the same names.



Actually we can delete one point in a face of a polyhedron. and expand it to get a planar graph (after deleting, the polyhedron $\cong \mathbb{R}^2$)

Conversely, one can do one-point compactification on the unbounded face of a planar graph, make it $\cong S^2$ and get a polyhedron. (Euler characteristic is also defined for CW complexes)

Pf: We start with a spanning tree (V,T) of G.

Then $|T|=n-1$ and since there is no circuit. $l=1$.

$n-(n-1)+1=2$. checked.

Now draw remaining edges one at a time. Each time we decomposes a face into two. Because adding any

edge to a tree generates a circle. (why?)
This completes the proof. $\square$

Cor For any planar graph $G=(V,E)$, let n, m, l be those in the theorem, c be the number of connected components of G. then $n-m+l=1+c$.

Pf: Exercise. $\square$

Maximally connected planar graph

Q: For a given number of vertices, intuitively, the less # of edges we have, the more likely for the graph to be planar. Is there a threshold m_0 such that any graph with $m>m_0$ cannot be planar?

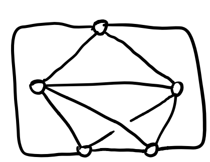
A: Yes.

Def A connected planar graph is <u>maximal</u> if adding any extra edge violate planarity.

Prop For a maximal connected planar graph with $n\geq 3$, $m=3n-6$, $l=2n-4$.

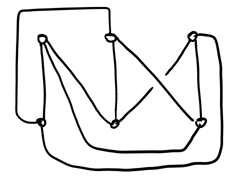
Pf: Claim that every face in such a graph is a triangle, i.e. has three boundary edges.
Otherwise one can certainly draw a diagonal on that face without ruining planarity.
So $m=\frac{3}{2}l$ and $n-m+l=n-\frac{3}{2}l+l=2 \Rightarrow \begin{cases} m=3n-6 \\ l=2n-4 \end{cases}$. $\square$

E.g ① K_5, the complet graph of order 5



$n=5$, $m=\binom{5}{2}=10$, $m>3n-6=9$
Does not have an embedding by Euler's relation

② $K_{3,3}$, the full 3+3 bipartite graph. $n=6$, $m=9$



$m \le 3n-6 = 12$, but still not a planar graph.

Problem arise in the (equivalent) definition of bipartite graph: every circuit has even length.

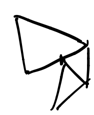
So "every graph with $m \le 3n-6$ is planar" is wrong!

Straight-line embedding

Def Given $\{a_i\}_{i=0}^{k} \subseteq \mathbb{R}^2$, we call $x=\sum_{i=0}^{k} t_i a_i$ a <u>convex combination</u> of a_i's if $t_i \ge 0 \in \mathbb{R}$ and $\sum_{i=0}^{k} t_i = 1$.

The <u>convex hull</u> of $\{a_i\}$ is the set of all convex combinations, namely $\{\sum_{i=0}^{k} t_i a_i \mid t_i \ge 0, \sum_{i=0}^{k} t_i = 1\}$

Let $G=(V,E)$ be a connected planar graph with a given embedding. We distinguish the boundary vertices with the interior vertices. More precisely, the boundaries of the unbounded face are called bondary edges whose vertices are called boundary vertices.



Def Let G be as above. A map $f: V \to \mathbb{R}^2$ is a <u>strictly convex combination mapping</u> if the images of the boundary vertices are vertices of a strictly convex polygon and for every interior vertex $u \in V$, $\exists\, t_{uv} > 0$, $\sum_v t_{uv} = 1$ such that $f(u) = \sum_v t_{uv} f(v)$, where the sum is taken over all neighbours of u. (Neighbours are those who connected directly with u)

Thm (Tutte's) Let G be a graph as above and $f: V \to \mathbb{R}^2$ be a strictly convex combination mapping. Then drawing straight edges between the images of the vertices gives a straight-line embedding.

Cor Every connected planar graph has a straight-line embedding.

Pf: Let $G=(V,E)$ be such a graph, with boundary and interior vertices distinguished by an embedding.

Denote the boundary vertices by $v_1, \dots, v_n$ and interior vertices by $u_1, \dots, u_k$.

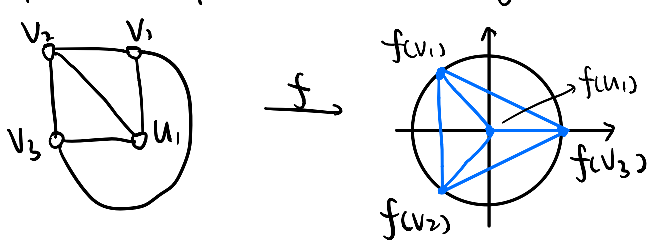
Let $f(v_i) = (\cos\frac{2\pi i}{n}, \sin\frac{2\pi i}{n})$, $f(u_j) = \frac{1}{d_j}\sum_v f(v)$, where the sum is taken over all neighbours of u_j and d_j is the degree of u_j. (# of neighbours)

Then we actually get k linear equations in k unknowns. Solving for $f(u_j)$'s, we constructed a strictly convex combination map.

By Tutte's theorem, drawing straight edges between these vertices gives a straight line embedding. □

Rmk: This is actually a "barycentric combination map". Every interior point is the barycenter of its neighbours.

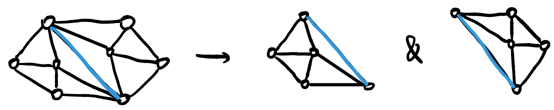
Simple example: The full graph of order 4



Proof of Tutte's theorem

We may assume all bounded faces of G are triangulars (have three edges), otherwise complete it and regard G as a subgraph.

We may also assume G does not have a separating edge, that is, an interior edge whose endpoints are all boundary points. Otherwise, replace G by its subgraphs defined by the separating edge.



Def A path is called interior if it consist of only interior vertices except possibly the first and the last.

lemma1 For every interior vertex u, every boundary vertex v, there is an interior path connecting them.

Pf: Strat with u, if none of neighbours of u are boundary vertex, we randomly choose one and look at its neighbours. Continue this fashion. Since there are only finitely many interior vertices, we finally reach a boundary vertex and find an interior path.

Now it remains to show that there is an interior path connecting two adjacent boundary vertices. Actually, since there is no separating edge, the neighbours of one of them suffices.

lemma2 let $h: \mathbb{R}^2 \to \mathbb{R}$, $h(x) = \langle x, p \rangle + c$ for some fixed $p \neq 0 \in \mathbb{R}^2$, $c \in \mathbb{R}$, then $h \circ f: V \to \mathbb{R}$ satisfies the maximal/minimal principle: If $h \circ f$ reach its maximum/minimum at an interior vertex, then $h \circ f$ is a constant on V.

Pf: Suppose $h\circ f$ take its maximum on an interior vertex u with $f(u)=\sum_v t_{uv} f(v)$, a convex combination

Observe that $h(f(u))=h(\sum_v t_{uv} f(v))=\sum_v t_{uv} h(f(v))$

Since $h\circ f(v)\le h\circ f(u)$ for all its neighbour v, it means

$h(f(u))=\sum_v t_{uv} h(f(v))\le\sum_v t_{uv} h(f(u))=h(f(u))$. The equality holds only if $h\circ f(v)=h\circ f(u)$ for all v.

So $h\circ f$ takes the same value for all neighbours of u.

Continue this fashion, by lemma 1, we will eventually reach all vertices. Thus $h\circ f$ is a constant.

The same for minimum.

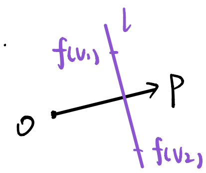
What we want to prove:

(1) All interior points are mapped into the interior of the polygon made by $\{f(v_i)\}$, $\{v_i\}$ the boundary points

(2) For any $(u_1, v_1)\ne(u_2, v_2)\in E$, the line segments $f(u_1)f(v_1)$, $f(u_2)f(v_2)$ do not intersect.

Once (1) (2) are proved, we are done.

(1) Fix a boundary edge (v_1, v_2). Consider the line l determined by $f(v_1), f(v_2)\in\mathbb{R}^2$. Choose suitable $p\in\mathbb{R}^2$, $c\in\mathbb{R}$ s.t. $h^{-1}(0)=l$.



(why can we do this? $h^{-1}(0)=\{x\in\mathbb{R}^2 \mid \langle x, p\rangle=-c\}$ is a straight line. Take p to be perpendicular to l, then $\langle x, p\rangle$ is a constant on l. Let $c=-\langle x, p\rangle$.)

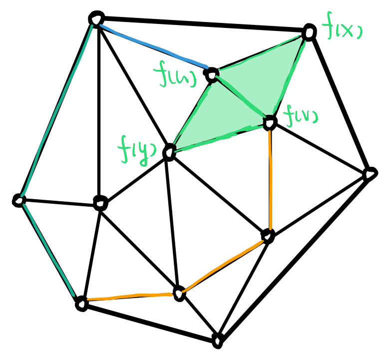
Now since images of the boundary vertices lie on a strictly convex polygon, they are all at one side of l.

Thus $h\circ f(v)<0$ (or >0, WMA "<") for all other boundary vertices. Now if $h\circ f(u)\ge 0$ for some interior u, then $h\circ f$ will take maximum at u, forcing $h\circ f$ to be constant by lemma 2, but this is not true.

So $h \circ f(u) < 0$ for all interior vertices u. This means all $f(u)$ lie in the same side of l as other $f(v)$, images of boundary vertices.

Do this for all boundary edges, we conclude that all interior $f(u)$ lie in the interior of the polygon.

For (2), we need another lemma. Before that, note that every interior edge (u, v) corresponds to exactly two triangular faces $(u v x)$, $(u v y)$. Then so do $(f(u), f(v))$.



Lemma 3 The two vertices $f(x)$, $f(y)$ forming triangles with $f(u)$, $f(v)$, lie on different sides of the line $f(u)f(v)$.

Pf: Choose suitable p, c such that $h^{-1}(0)$ passes through $f(u)$, $f(v)$. Suppose $h(f(x)) < 0$, we'll show $h(f(y)) > 0$.

Since $h(f(u)) = 0$, $h(f(y)) > 0$, there is a neighbour u' of u such that $h(f(u')) < 0$. Now $h(f(u')) < h(f(u))$, do the same for u' and so on, until we reach the boundary, by a path along which the function $h \circ f$ strictly increase.

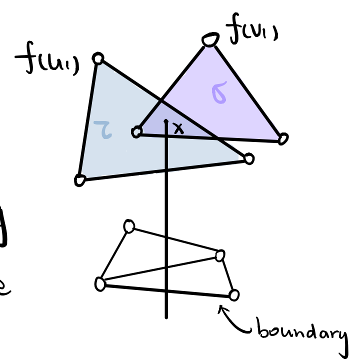
Similarly, there is another such strictly-increasing path strated with v.

The two paths, together with (u, v) and serveral possible boundary edges, form a cycle whose vertices have non-negative function values. The images of these vertices form a polygon. By the convexity of f, $f(y)$ is in the convex hull of the polygon. By minimal principle, we have $h(f(y)) > 0$, which implies that $f(y)$ is at the opposite side.

why not $f(x)=0$? Indeed, we may choose a sequence of triangular faces strated from wvx, until we reach a boundary edge. In this sequence, any two contignous triangles share one edge. By (1), image of the last Δ is non-degenerate. This implies the second last one is also non-degenerate by the strict convexity, and therefore implies $\Delta(f(x)\, f(w)\, f(v))$ is non-degenerate.

Now prove (2): We'll show that any different triangles $\Delta(f(u_1)\, f(u_2)\, f(u_3))$, $\Delta(f(v_1)\, f(v_2)\, f(v_3))$, have disjoint interior.

Suppose we have two such images, σ and τ with a common point x.



Draw a half line l emanating from x avoiding any vertex in V. Then it intersect one of the edges of τ.

If that is not boundary edge, then by lemma 3, it is an edge of another triangle lying on opposite side of τ, and l will intersect one of the other two edges of that triangle. Contihue doing this until reach the bondary and get a sequence of triangles, strated with τ, ended with some boundary triangle β.

Do the same for σ, get another seq of triangles, strated with σ, ended with the same β.

Now we go backward, strat at β along the half-line l, follow the same rule. Since β is a known triangle, every step backward unambiguonsly determines a preceeding triangle.

It follows that $\tau=\sigma$, completing the proof. $\square$