15-853:Algorithms in the Real World

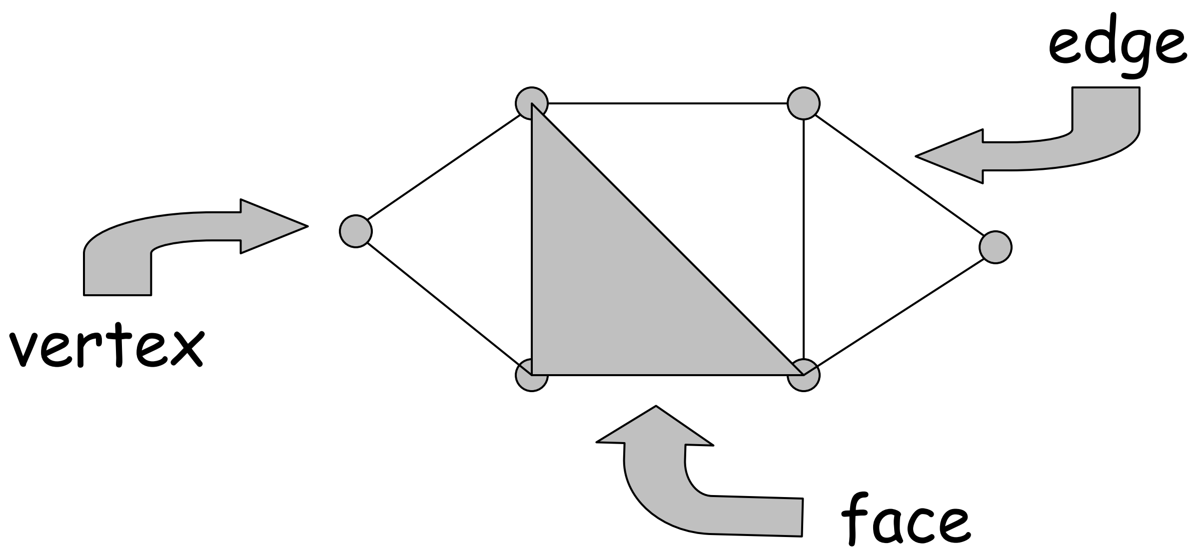
Planar Separators I & II

- Definitions
- Separators of Trees
- Planar Separator Theorem

Planar Graphs

Definition: A graph is planar if it can be embedded in the plane, i.e., drawn in the plane so that no two edges intersect.

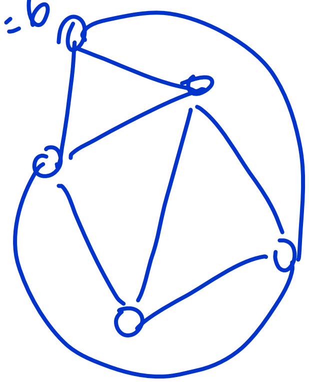
(equivalently: embedded on a sphere)



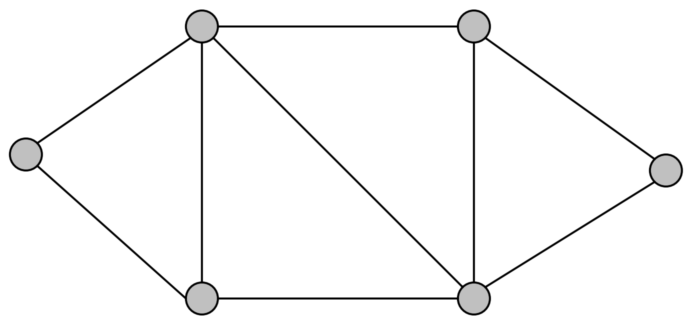
Euler's Formula

Theorem: For any spherical polyhedron with V vertices, E edges, and F faces,

V=5
E=9
F=6



$$V - E + F = 2.$$



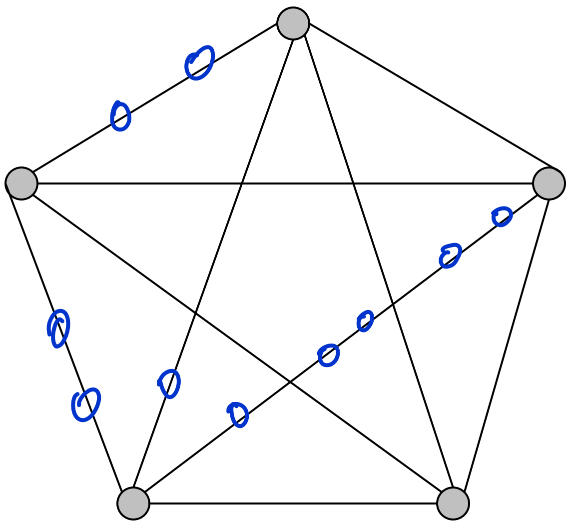
V = 6
E = 9
F = 5

Corollary: If a graph is planar then $E \leq 3(V-2)$
planar graph with n nodes has O(n) edges.

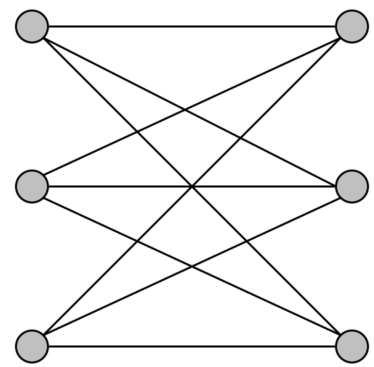
(Use $2E \geq 3F$.)

Kuratowski's Theorem

Theorem: A graph is planar if and only if it has no subgraph homeomorphic to K_5 or $K_{3,3}$.



K_5

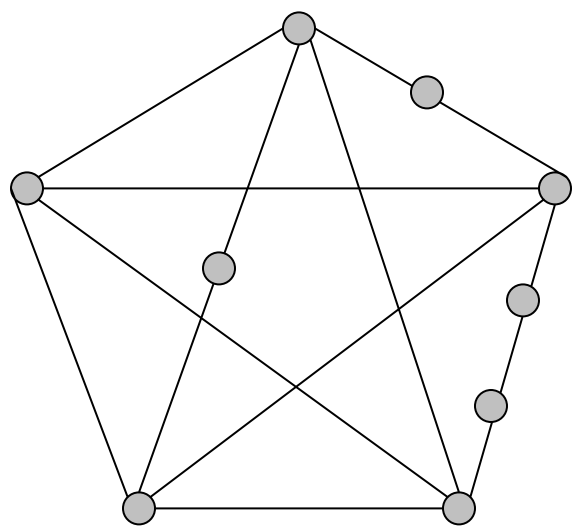


$K_{3,3}$

"forbidden subgraphs" or "excluded minors"

Homeomorphs

Definition: Two graphs are homeomorphic if both can be obtained from the same graph G by replacing edges with paths of length 2.



A homeomorph of K_5

Algorithms for Planar Graphs

Ungar 1951: an $O(\sqrt{n} \log n)$ separator theorem for planar graphs.

Hopcroft-Tarjan 1973: Algorithm for determining if an n-node graph is planar, and, if so, finding a planar embedding, all in $O(n)$ time. (Based on depth-first search.)

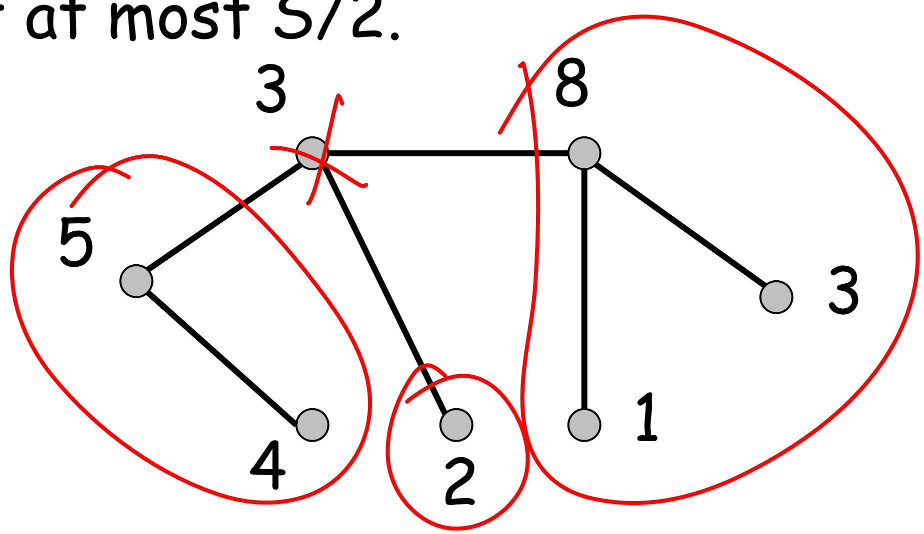
Algorithms for Planar Graphs

Lipton-Tarjan 1977: Proof that planar graphs have an $O(\sqrt{n})$-vertex separator theorem, and an algorithm to find such a separator.

Lipton-Rose-Tarjan 1979: Proof that nested-dissection produces Gaussian elimination orders for planar graphs with $O(n \log n)$ fill.

Separators of Trees

Theorem: Suppose that each node v in a tree T has a non-negative weight w(v), and the sum of the weights of the nodes is S.
Then there is a single node whose removal (together with its incident edges) separates the graph into at least two components, each with weight at most S/2.

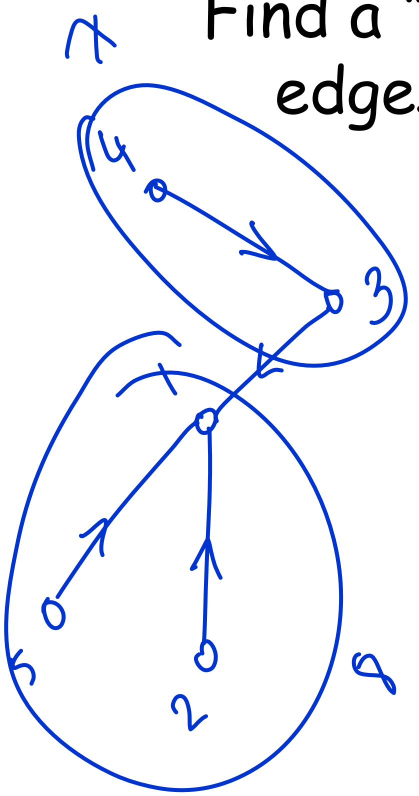


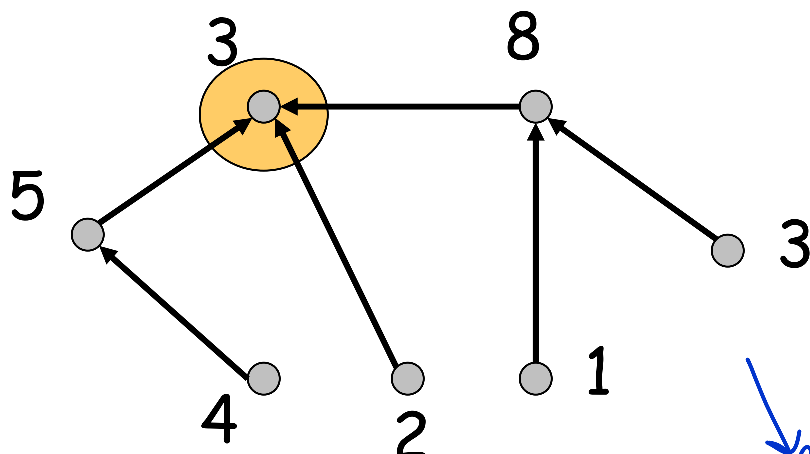
S = 26

Proof of Theorem:

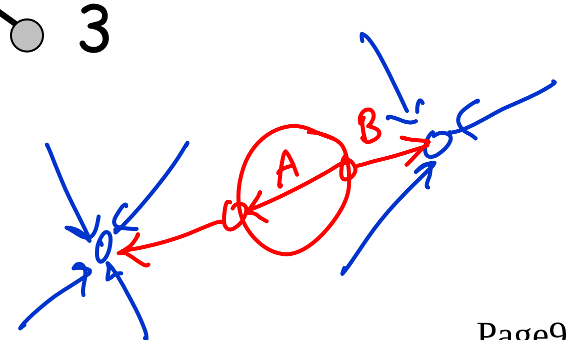
Direct each edge towards greater weight. Resolve ties arbitrarily.

Find a "terminal" vertex – one with no outgoing edges. This vertex is a separator.



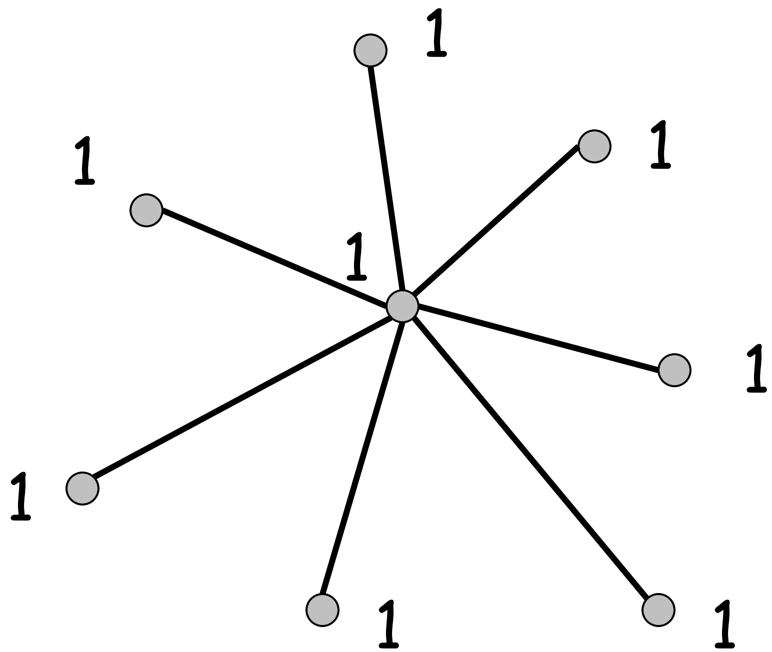


S = 26



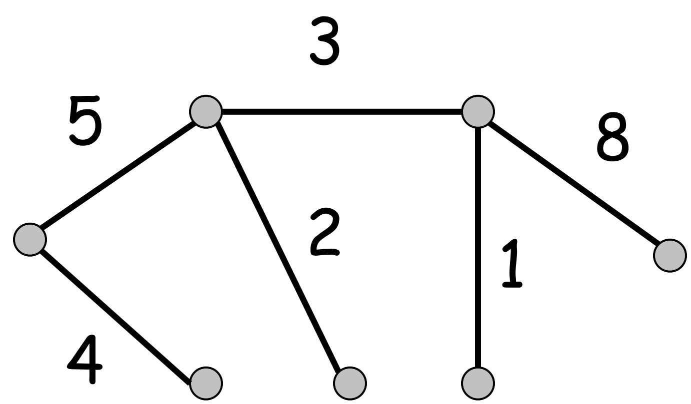
Observation

There is no corresponding theorem for edge separators.



Weighted Edges

Theorem: Suppose that each edge e in a $\boxed{\text{degree-D}}$ tree T has a non-negative weight w(e), and the sum of the weights of the edges is S.
Then there is a single edge whose removal separates the graph into two components, each with weight at most (1-1/D)S.



S = 23

$D = 3 \Rightarrow$ each component wt $\leq \frac{2}{3} \cdot S$

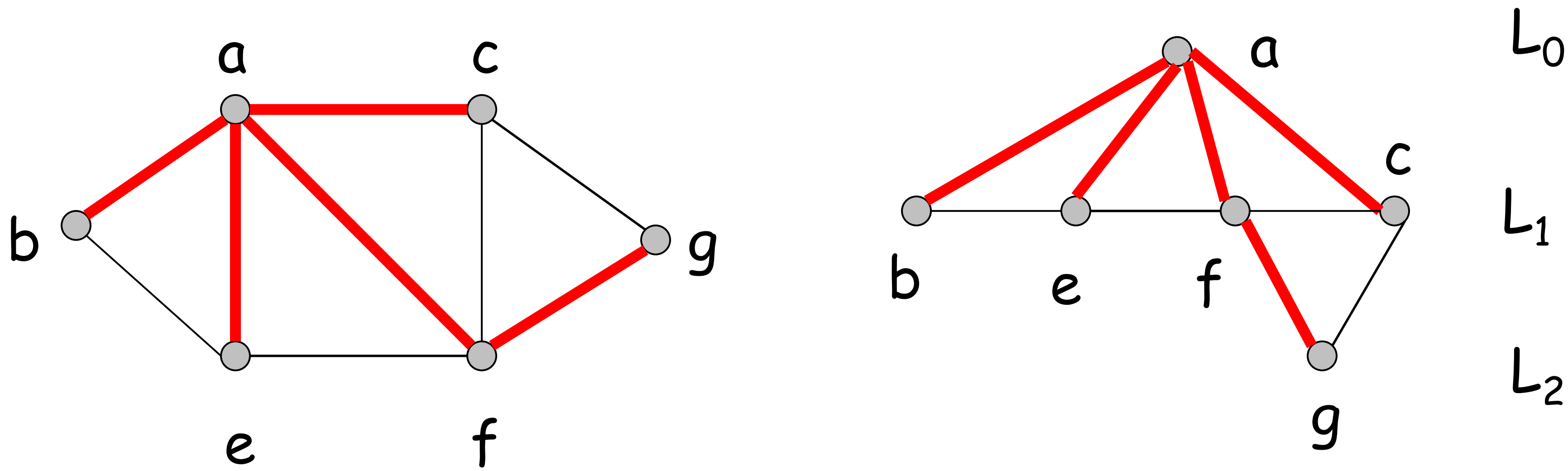
Proof: Exercise.

Planar Separator Theorem

Theorem (Lipton-Tarjan 1977): The class of planar graphs has a (2/3,4) $\sqrt{n}$ vertex separator theorem. Furthermore, such a separator can be found in linear time.

Planar Separator Algorithm

Starting at an arbitrary vertex, find a breadth-first spanning tree of G. Let L_i denote the i'th level in the tree, and let d denote the number of levels.



Observe that each level of tree separates nodes above from nodes below.

Algorithm CUTSHALLOW

Theorem: Suppose a connected planar graph G has a spanning tree whose depth is bounded by d. Then the graph has a 1/3-2/3-vertex separator of size at most 2d+1.

Proof later.

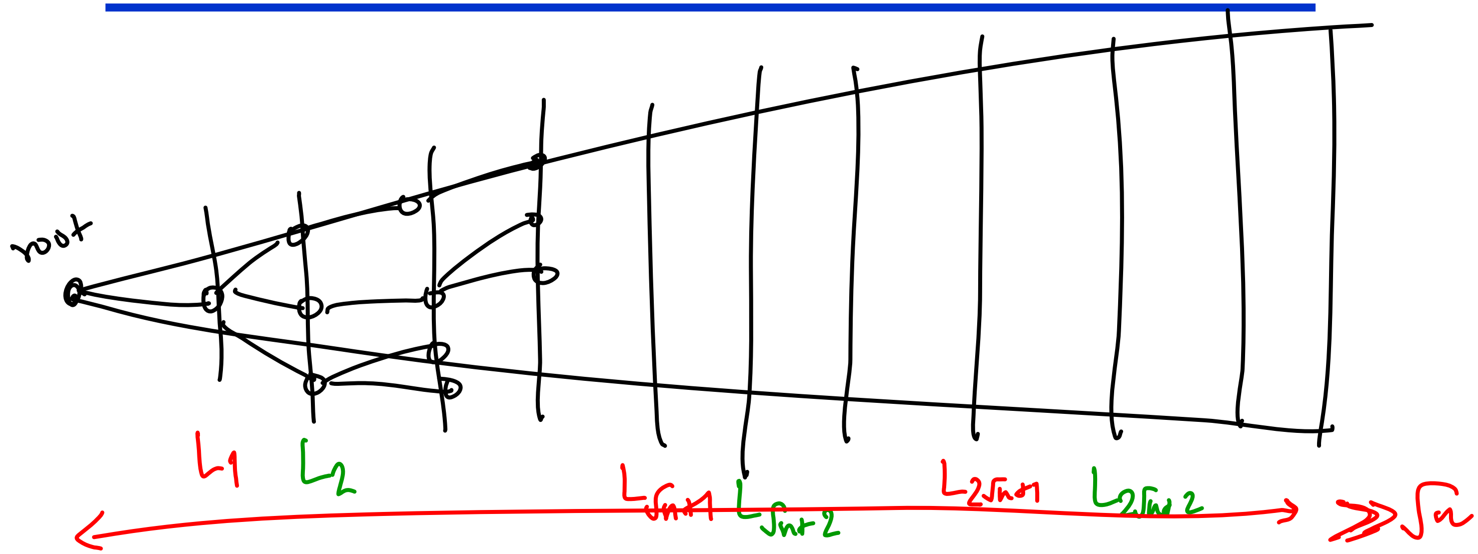
What if G is not connected?

If there is a connected component of size between n/3 and 2n/3, we have a separator of size 0.

If all components have size less than n/3, we have a separator of size 0.

Otherwise, separate largest component (of size >= 2n/3).

Breadth First Search Tree



Suppose I ~~...~~ am given a graph with depth $\gg \sqrt{n}$

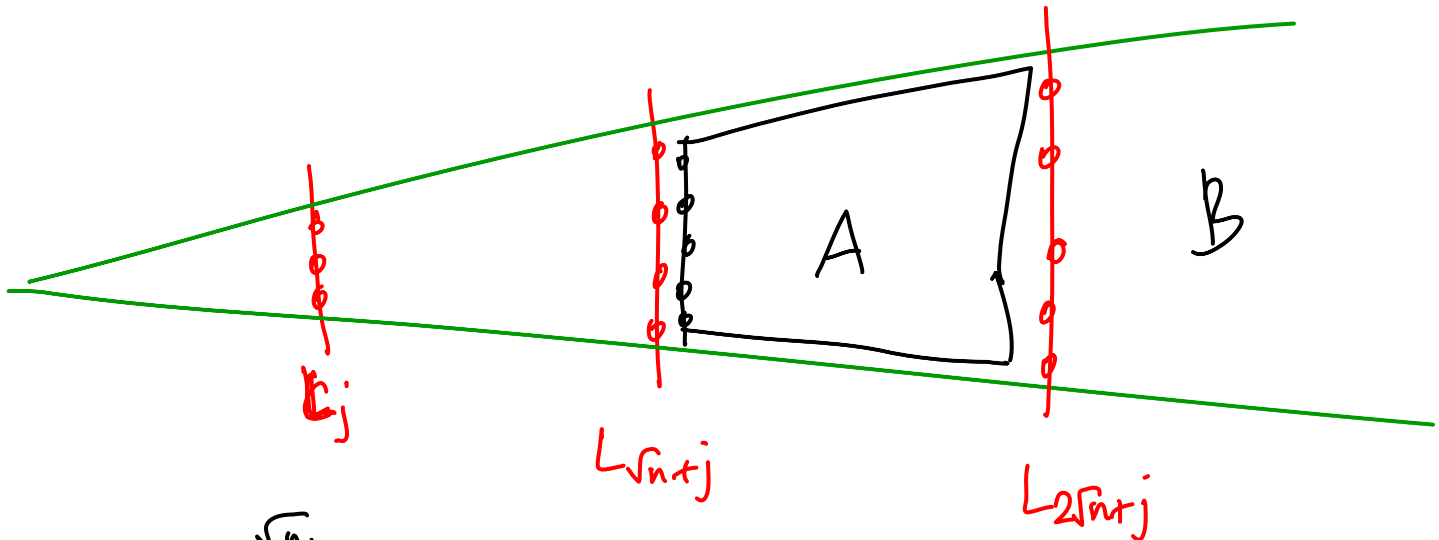
$C_1 = L_1 \cup L_{\sqrt{n}+1} \cup L_{2\sqrt{n}+1} \ldots\ldots$ $\quad C_0\ C_1, C_2 \ldots C_{\sqrt{n}-1}$

$C_2 = L_2 \cup L_{\sqrt{n}+2} \cup L_{2\sqrt{n}+2}$

$C_j = L_j \cup L_{\sqrt{n}+j} \cup L_{2\sqrt{n}+j} \ldots$

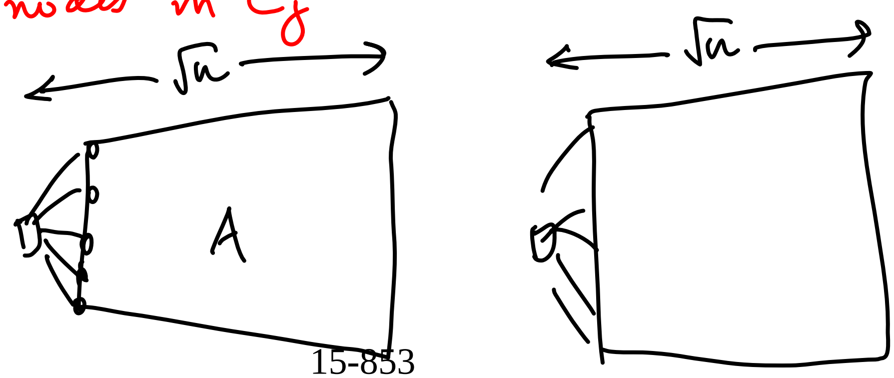
$$|C_0 \cup C_1 \cdots \cup C_{\sqrt{n}-1}| = n$$

$$\Rightarrow \exists C_j \text{ s.t. } |C_j| \le \frac{n}{\sqrt{n}}$$



$\sqrt{n}$

Delete all nodes in C_j

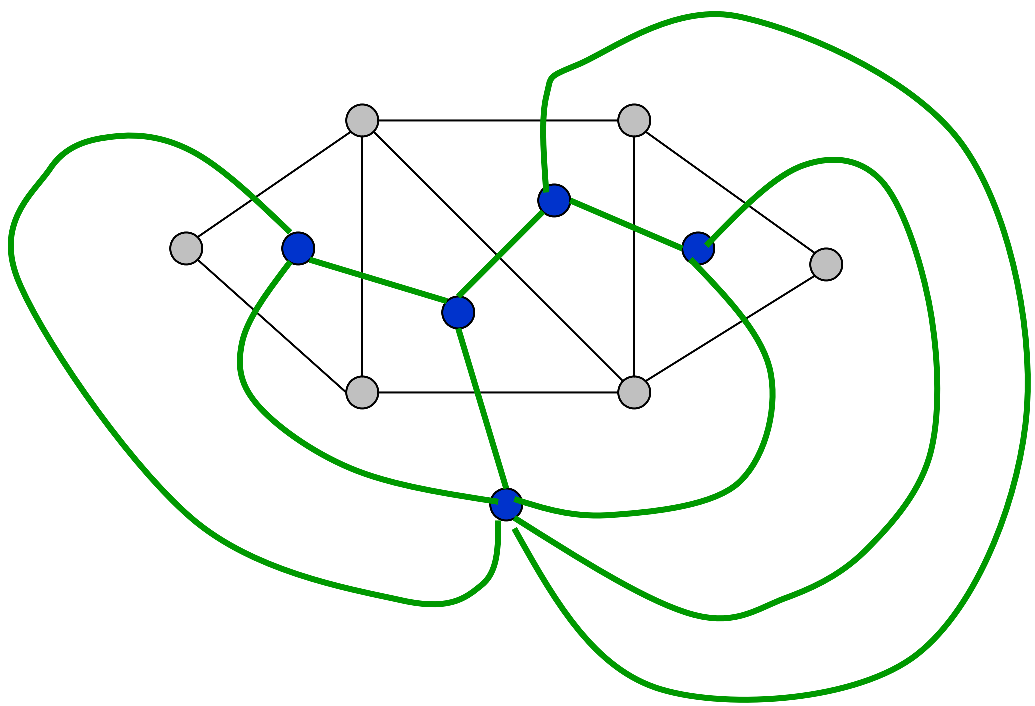


Thm: if we are given an arbitrary planar graph, G we can delete $\sqrt{n}$ nodes to obtain a planar graph, each of whose components has depth $d \leq \sqrt{n}$.

$\Rightarrow$ by CutShallow, I can delete further $2d+1$
$\leq 2\sqrt{n}+1$
nodes and obtain a $1/3$-$2/3$ separator for the planar graph G.

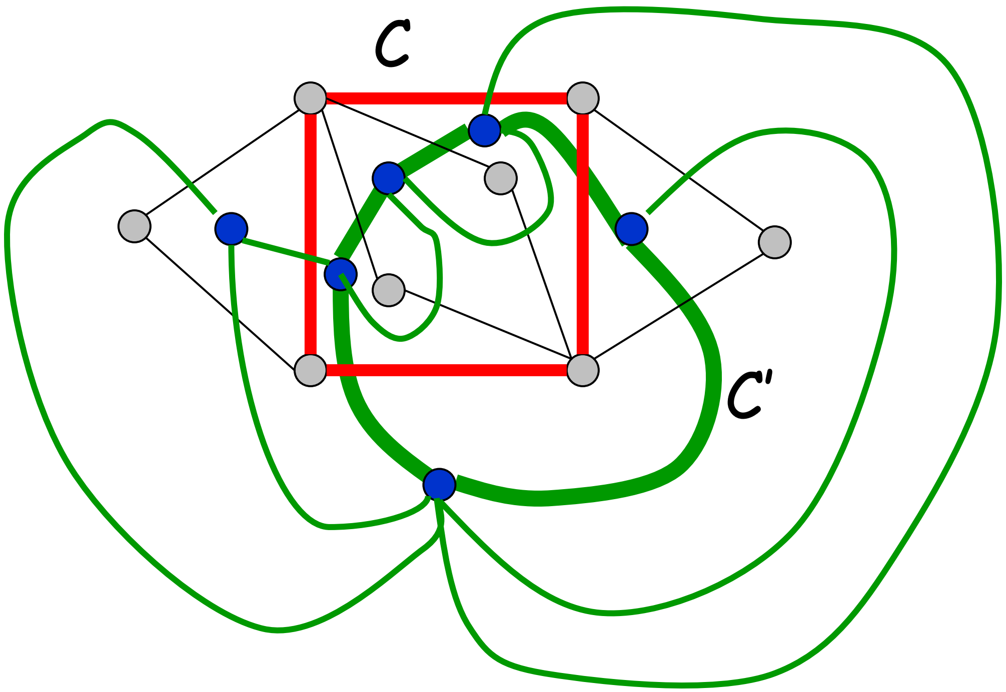
Dual Graph

In the dual of a plane graph G (planar graph embedded in the plane), there is a node for each face of G, and an arc between any two faces that share an edge in G. The dual graph is also planar.



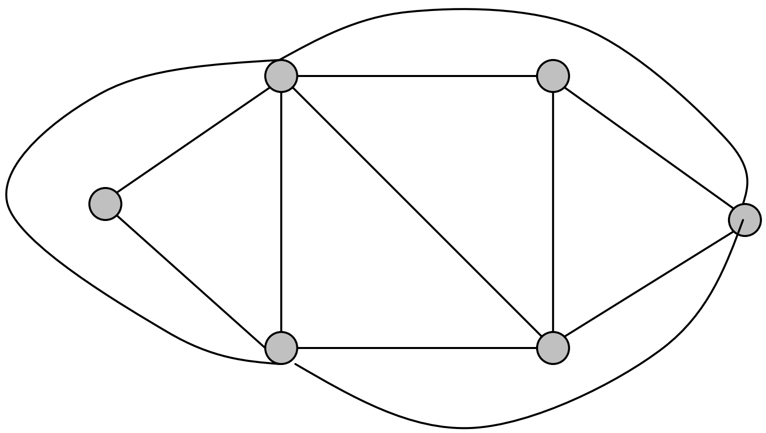
Cycles

A cycle C in G vertex-separates (or edge-separates) the vertices and edges inside C from those outside. Similarly, a cycle C' in the dual of G edge-separates the vertices of G inside C' from those outside.



Triangulation

In a triangulated plane graph, every face (including the external face) has three sides.



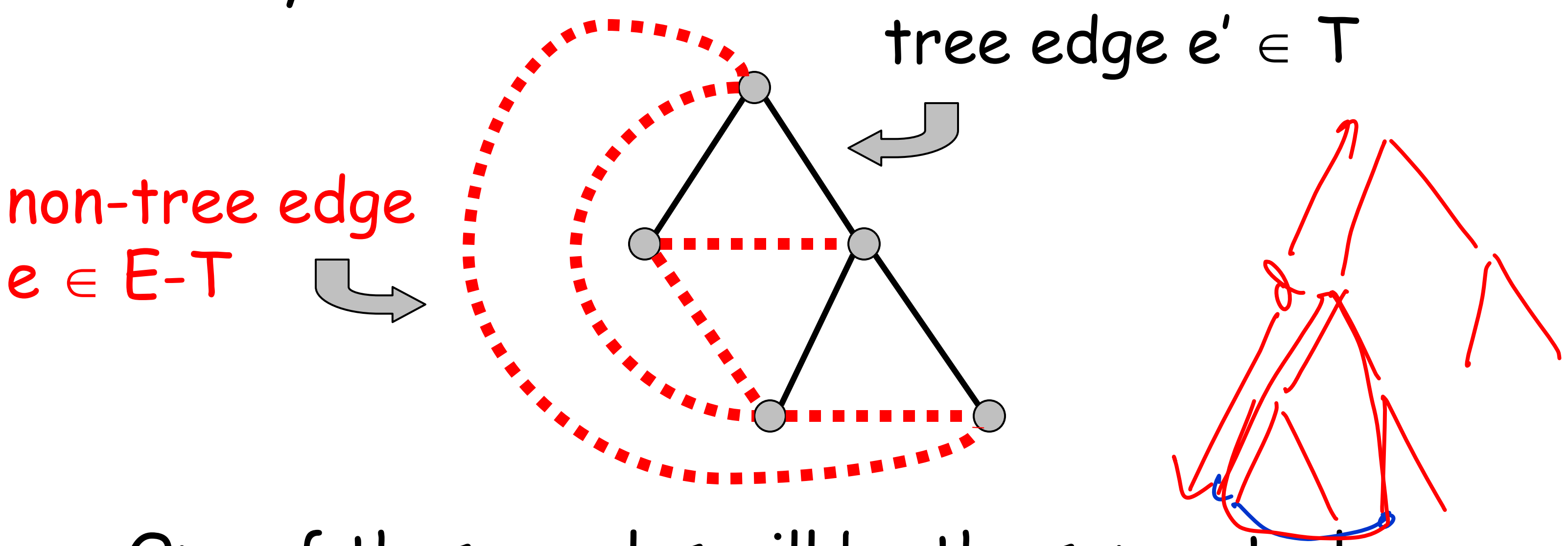
Theorem: Any plane graph can be triangulated by adding edges.

(Use $E \leq 3 (V-2)$.)

Algorithm CUTSHALLOW

Start with any depth d spanning tree T of G. (T need not be a breadth-first search tree.) Assume G has been triangulated.

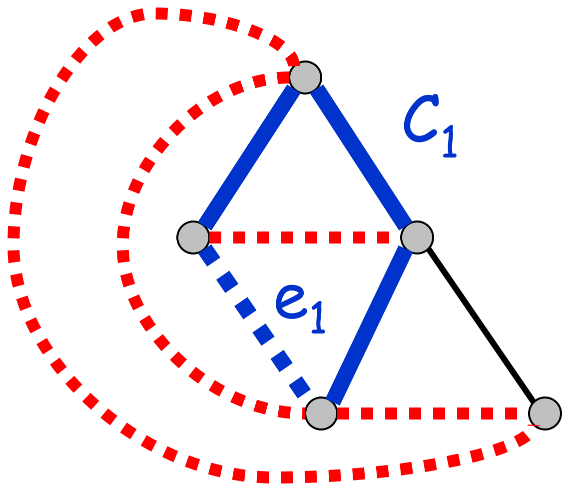
Observe that adding any non-tree edge e ∈ E-T to T creates a cycle in G.



One of these cycles will be the separator!

Detour: Cycle Basis

Let e_1, e_2, …, e_k denote the non-tree edges.
Let C_i denote the cycle induced by e_i.



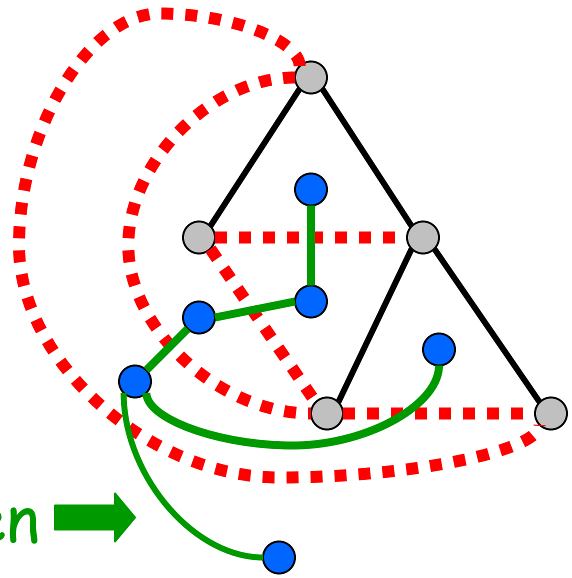
Let $C_1 \oplus C_2$ denote $(C_1 \cup C_2) - (C_1 \cap C_2)$, i.e., symmetric difference.

Theorem: Any cycle C in G can be written as $C_{i1} \oplus C_{i2} \oplus \cdots \oplus C_{ij}$ where e_{i1}, e_{i2}, …, e_{ij} are the non-tree edges in C.

Spanning Tree of Dual Graph

Let A denote the set of arcs in the dual graph D that cross non-tree edges of G.

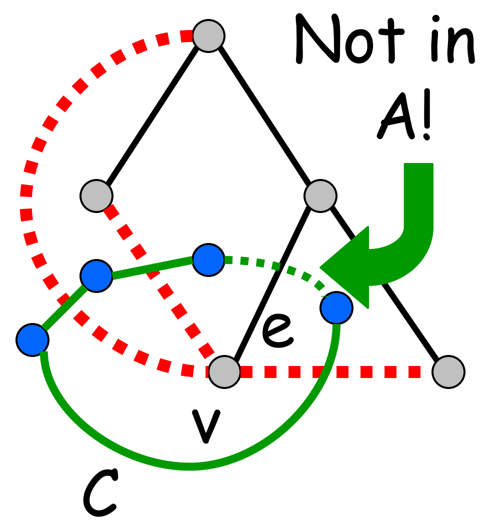
A in green



Spanning Tree of Dual Graph

Claim: A is a spanning tree of D.

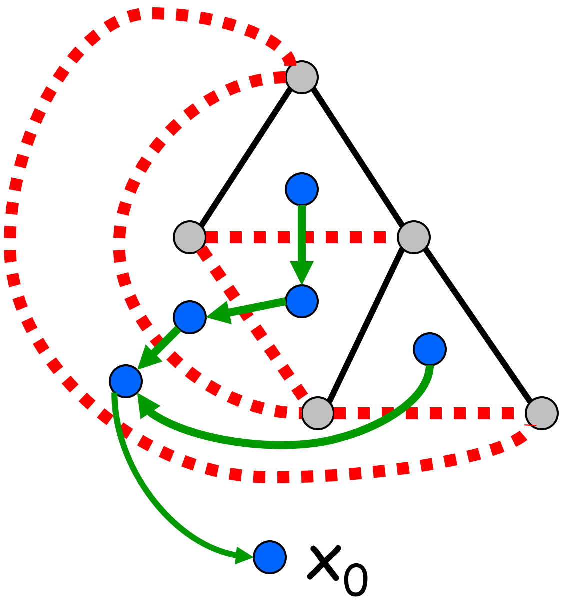
Proof:



1) A is acyclic -- Any cycle C in A would enclose some vertex v of G (since edges of E-T cross arcs of C). C also corresponds to an edge separator of G. But since T spans G, the separator would have to include an edge e $\in$ T, so C can't be made of only arcs of A, a contradiction.

2) A is spanning: there is a path in A between any two nodes of D -- because T is acyclic.

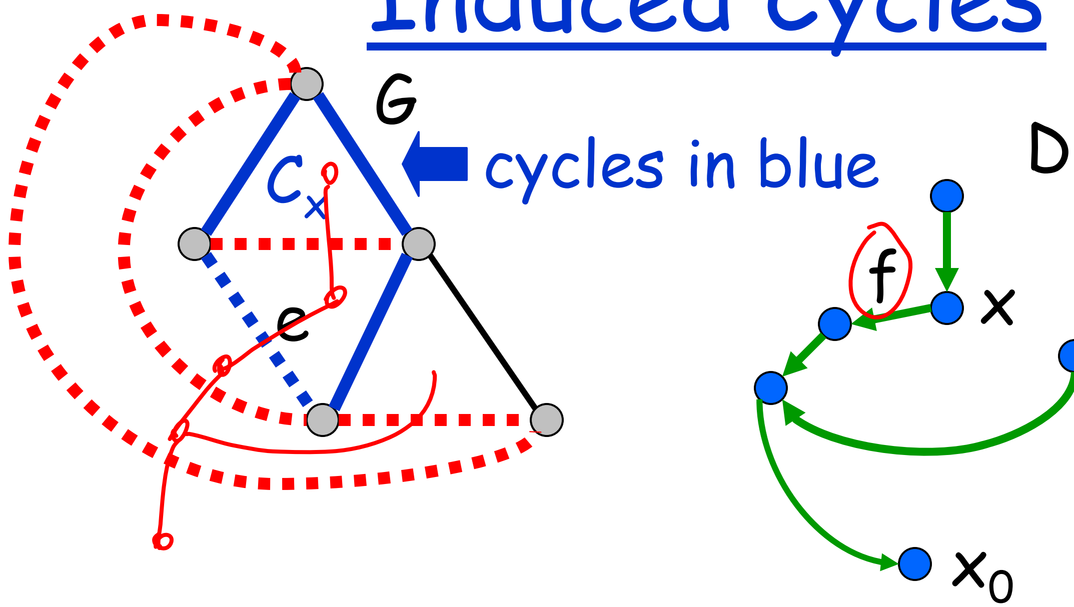
Rooting the Spanning Tree



Pick an arbitrary degree-1 node of A, call it x_0 and make it the root of A, directing arcs towards A.

For any cycle in G, call side containing x_0 the "outside".

Induced Cycles

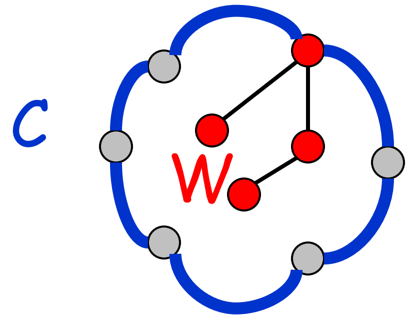


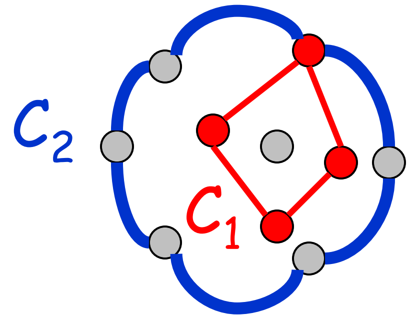
Suppose x is the child end of arc f $\in A$, which crosses edge e $\in$ E-T. Adding e to T induces cycle C_x, of depth at most 2d+1.

We say that C_x is <u>the cycle induced by x</u>.
Every node of D except x_0 induces a cycle in G.

Containment and Enclosure

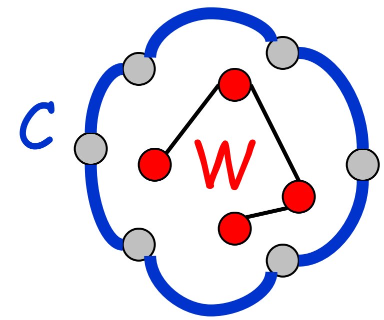
Given $W \subset V$, we say that cycle C in G **contains** W if every vertex in W is either inside or on C. Note that if the vertices on cycle C_1 are contained in cycle C_2, then any vertex inside C_1 is also inside C_2.





vertices of W shown red

Similarly, C **encloses** W (here W can be a set of vertices in G or a set of nodes in D) if all vertices of W are inside of C.

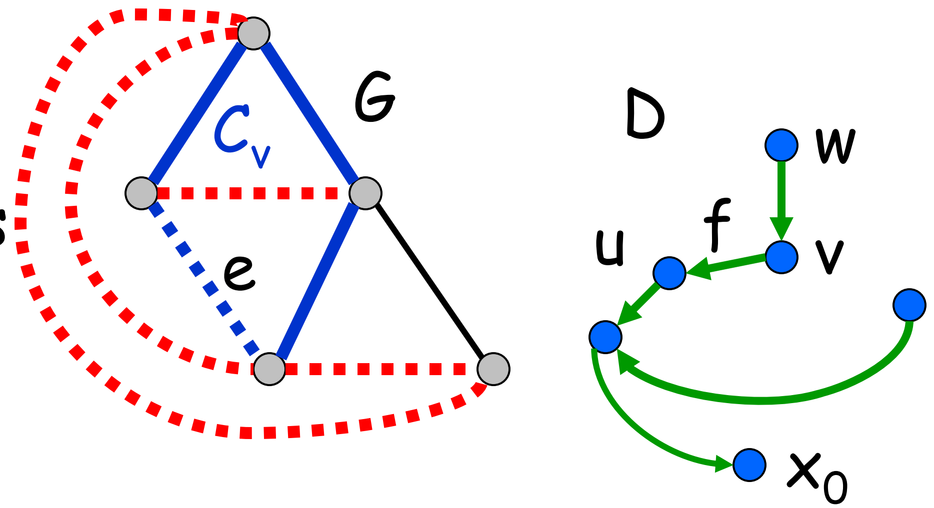


Lemma 1

Lemma 1: Suppose C_v is the cycle induced by node v. Then all the nodes in the subtree of A rooted at v are inside C_v.

Proof: Let u be the parent of v in A. Let f = {v,u} be the arc from v to u, and let e be the edge in E-T that f crosses. Edge e induces cycle C_v in G, and u and v are on different sides of C_v.

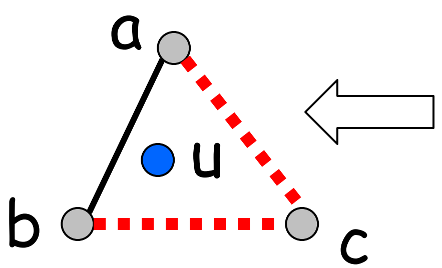
Suppose w is a descendant of v. There is a path from face v to face w that doesn't cross f or any edge of T. Hence w is on the same side of C_v as v. Similarly, x_0 is on the same side of C_v as u.



Lemma 2

Lemma: The cycle induced by a node of D contains the cycle induced by any one of its children. Moreover, the cycles induced by siblings do not enclose any common vertex.

Proof: Suppose u is neither a leaf nor the root of A, and corresponds to a face {a,b,c} of G. Since G is triangulated, u can have either one or two children.



Case 1: u has only one child v.

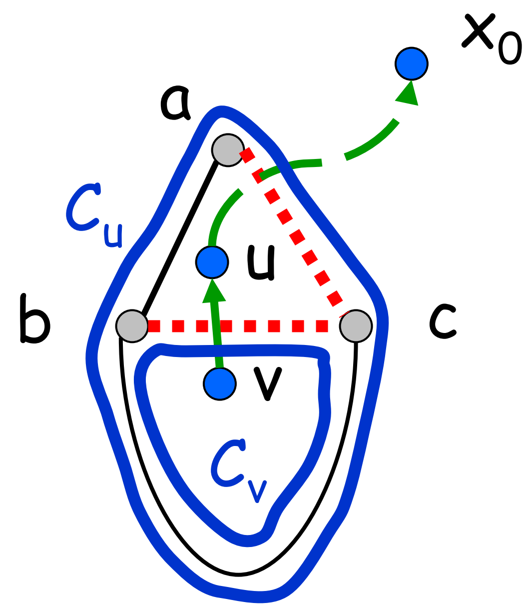
Case 2: u has two children, v and w.

Case 1 (a)

Case 1: u has one child v. Assume w.l.o.g. that arc (v,u) crosses edge {b,c}, and that {a,b} $\in$ T, and {b,c},{c,a} $\in$ E-T.

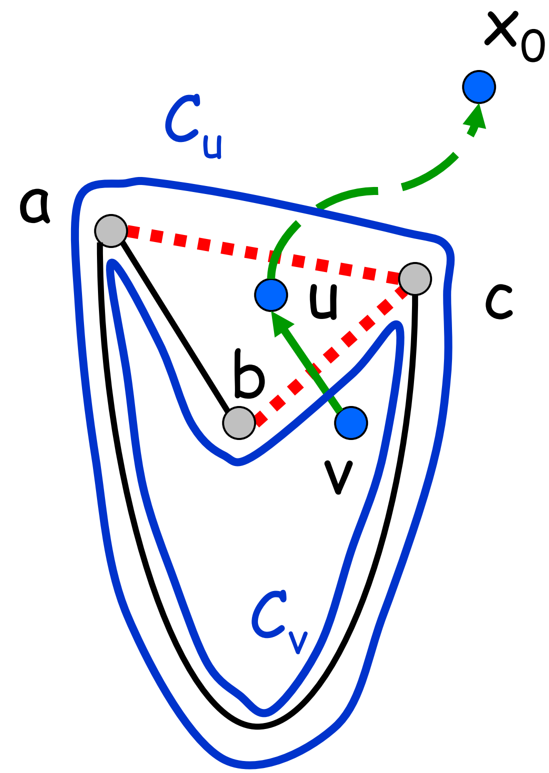
Case 1 (a): a does not lie on the cycle C_v.

Node u lies outside C_v, so vertex a must also lie outside C_v. Hence, C_u contains C_v and the two cycles enclose the same set of vertices of G.



Case 1 (b)

Case 1 (b): a lies on the cycle C_v.



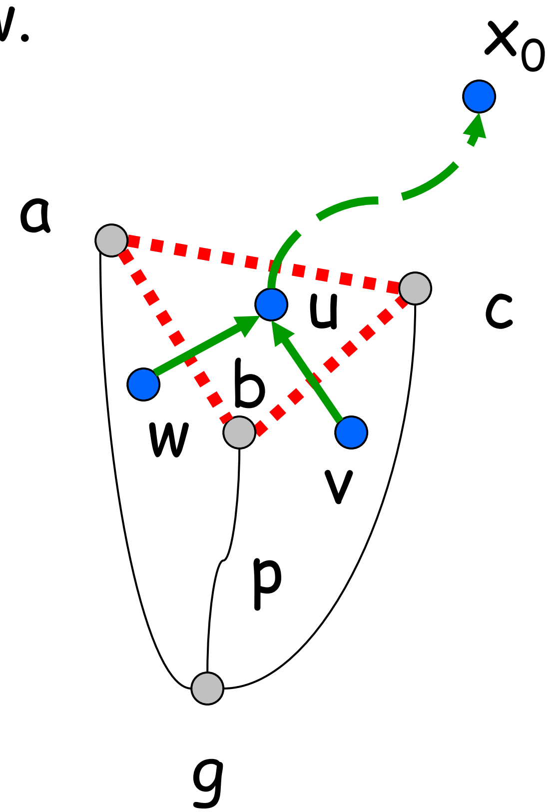
The root, x_0, is outside of C_u and u is inside of C_u. By Lemma 1, v is also inside C_u. Hence, b is inside C_u, and C_u contains C_v.

Note that C_u encloses one more vertex than C_v, b.

Case 2

Case 2: Node u has two children, v and w.

By Lemma 1, neither v nor w can be on the same side of C_u as the root x_0. Hence, b is contained in C_u.

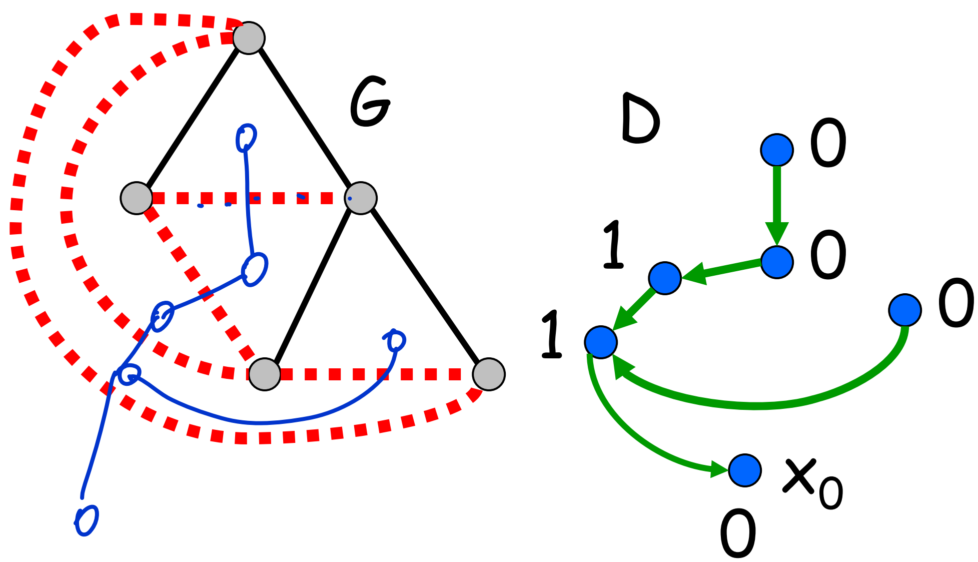


Since T is a spanning tree of G, there is a unique shortest path p in T from b to the cycle C_u, intersecting C_u at some vertex g. Path p is contained in C_u, and has length at most 2d.

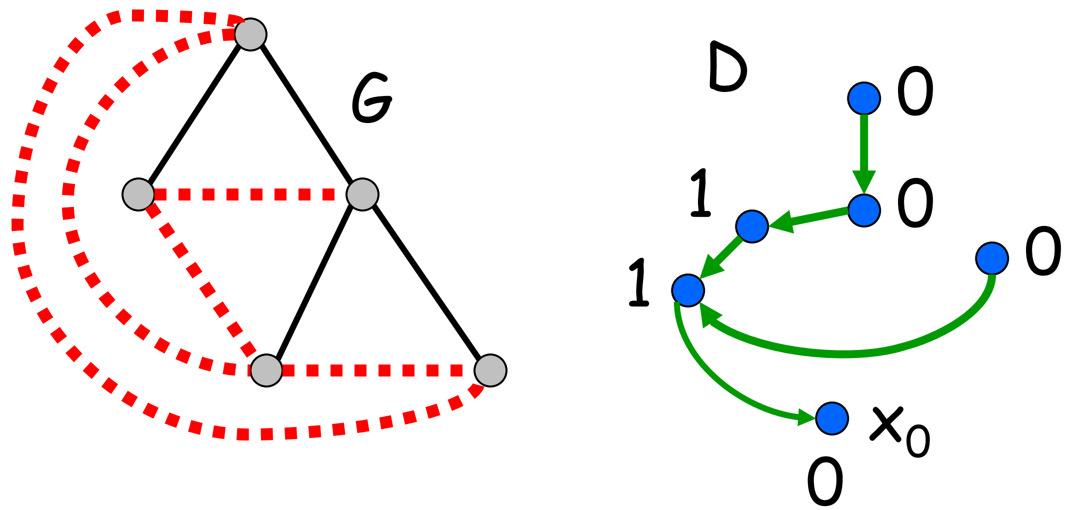
Both C_v and C_w are contained in C_u, and because G is planar, G partitions the vertices enclosed in C_u. C_u also encloses the vertices (except g) on p.

Assigning Weights to Nodes

Lemma: It is possible to assign a non-negative number $W(u)$ to each node u of D (except x_0) such that the number of vertices enclosed by the cycle induced by u equals the sum of the weights of the nodes in the subtree rooted at u. The weight of x_0 is defined to be 0. Moreover, the weight of a node is bounded by 2d.



Algorithm for Assigning Weights



Leaves and x_0 are assigned weight 0.

Let u be an internal node of A. From proof of Lemma 2:

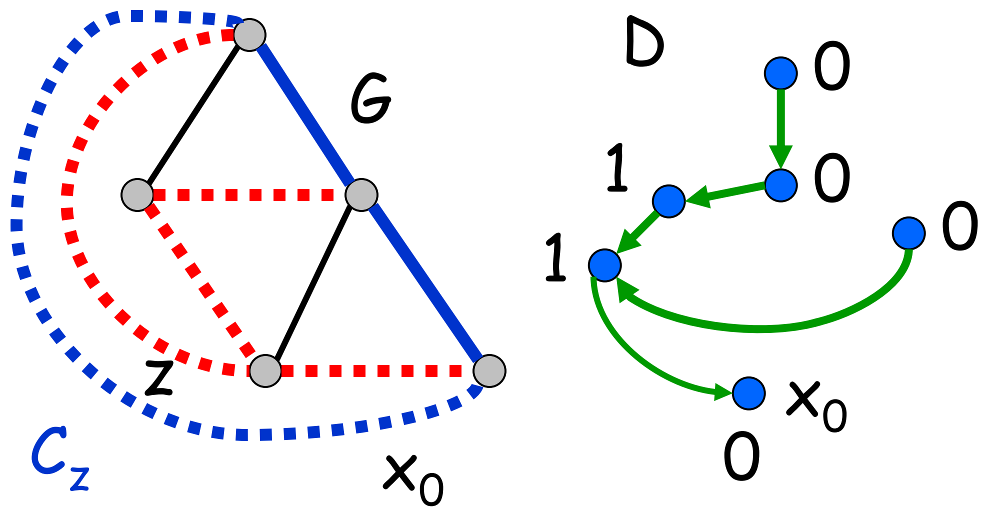
Case 1 (a): W(u) = 0; C_u encloses no more vertices than C_v.
Case 1 (b): W(u) = 1; C_u encloses one more vertex than C_v.
Case 2: W(u) = length of path p (at most 2d)

Total Tree Weight

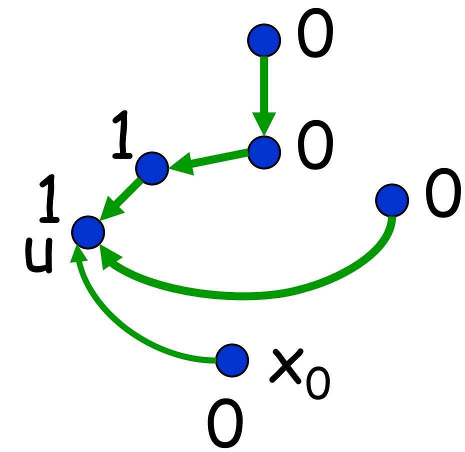
Lemma 4: The sum of the weights of the nodes of D is |V|-3, where G = (V,E).

Proof: Since $W(x_0) = 0$, the sum of the weights equals the number of vertices inside the cycle induced by the single child z of x_0.

The cycle C_z is a triangle corresponding to the face x_0. Hence, all vertices but the three on the triangle are enclosed by C_z.



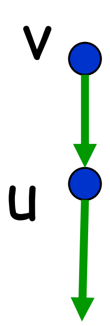
Finding a Separator



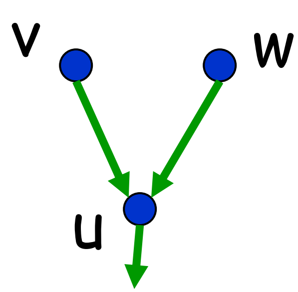
Redirect arcs toward greater weight.

Find a single-node (1/3,2/3)-separator of the tree *A* (a terminal node u).

Even though u is the separator, C_u might enclose more than 2n/3 vertices, because of its own weight W(u). If C_u encloses 2n/3 or fewer, it is the separator. Otherwise, two cases to consider: u has one child or u has two children in *A*. The rest is bookkeeping.



C_v is the separator



Whichever of C_v and C_w encloses more vertices is the separator