Graph Theory  MATH 442  Term 2  2018-09-11

CPSC 536E ~~topics~~ topics

matrix representation of graph ① computing path

② Google's page rank  how connected is the graph?

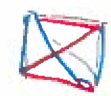
③ useful for graph drawing  eigenvalue

major topics:

TODO: ① spectral graph theory

~~② Ramsey's theory~~ Complete graph: color edge red or blue, smallest number of vertices.

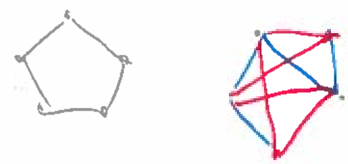
clique: → 



$R(3,3) = 3$ = size of smallest complete graph so that for all red/blue colorings the exists either a red 3-clique or a blue 3-clique.

$R(4,4) = 18$

$R(5,5) = 43$



③ Random walks on graphs

Give a graph, how many perfect matchings are there?

① ④ planar graphs

characterization  drawn on a plane without edges crossing.

Pointers for things to look at:

To get started with planar graphs

① Things for next time:

things about planar graphs

how to tell?

algorithms for planar detection

② graph drawing

② GRAPH DRAWING / 4 things interested in

ALGORITHMS FOR THE VISUALIZATION OF GRAPHS,

italian GIUSEPPE DI BATTISTA

Australian PETER EADES

italian ROBERTO TAMASSIA

Greek IOANNIS G. TOLLIS

③ Graph Theory with Applications

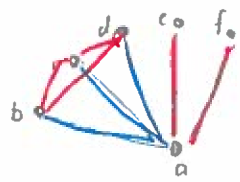
J.A. Bondy  U.S.R. Murty.

read chapter on planar graphs

Ramsey number

# Proof that $R(3,3) = 6$

In a complete graph ($K_6$), for node a adjacent to all other 5 nodes, color them red or blue.




No matter how you color it, at least 3 edges are colored with the same color (either red or blue).

WLOG, let's say a is connected with b, c, d colored blue.

To avoid a clique with a, bc, cd, bd has to be colored red. In this way, b, c, d form a red 3-clique.

Thus, $K_6$ must have either a red 3-clique or a blue 3-clique. $\square$

Bondy and Murty

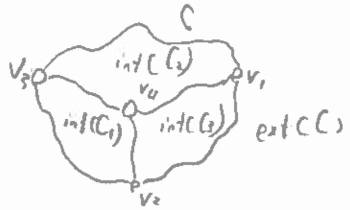
[1.1] P5 : planar graph: A graph which can be drawn in the plane in such a way that edges meet only at points corresponding to their common ends is called a planar graph, and such a drawing is called a planar embedding of the graph.

[10.1] P243 planar graph: A graph is said to be embeddable in the plane, or planar, if it can be drawn in the plane so that its edges intersect only at their ends.

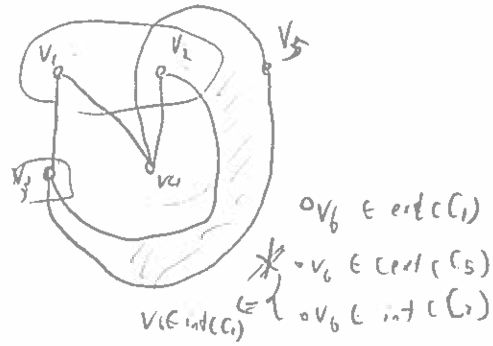
[10.1] P244 plane graph: we often refer to a planar embedding $\widetilde{G}$ of a planar graph $G$ as a plane graph; and we call the vertices of $\widetilde{G}$ points and its edges lines.

[10.1] P245 THE JORDAN CURVE THEOREM: Any simple closed curve $C$ in the plane partitions the rest of the plane into two disjoint arcwise-connected open sets.
└ called interior $int(C)$ and exterior $ext(C)$.

[10.1] P245 $K_5$ is nonplanar:




$K_{3,3}$:




[10.1] P246 subdivision: Any graph derived from a graph $G$ by a sequence of edge subdivisions is called a subdivision of $G$ or a $G$-subdivision.

[10.1] P246. Proposition 10.3: A graph $G$ is planar if and only if every subdivision of $G$ is planar.

[10.1] P247 Theorem 10.4: A graph $G$ is embeddable on the plane if and only if it is embeddable on the sphere.


stereographic projection from z

[10.2] P250 Proposition 10.5: Let $G$ be a planar graph, and let $f$ be a face in some planar embedding of $G$. Then $G$ admits a planar embedding whose outer face has the same boundary as $f$.
Proof idea - consider an embedding $\widetilde{G}$ of $G$ on the sphere.

[10.2] P250 Theorem 10.6: THE JORDAN-SCHÖNFLIESS THEOREM ?
Any homeomorphism of a simple closed curve in the plane onto another simple closed curve can be extended to a homeomorphism of the plane.

[10.2] P251 Theorem 10.7: In a nonseparable plane graph other than $K_1$ or $K_2$, each face is bounded by a cycle.
? ?

[10.2] P252 dual: Given a plane graph $G$, one can define a second graph $G^*$ as follows. Corresponding to each face $f$ of $G$ there is a vertex $f^*$ of $G^*$, and corresponding to each edge $e$ there is an edge $e^*$ of $G^*$. Two vertices $f^*$ and $g^*$ are joined by the edge $e^*$ in $G^*$ iff their corresponding faces $f$ and $g$ are sparated by the edge $e$ in $G$. Observe that if $e$ is a cut edge of $G$, then $f = g$, so $e^*$ is a loop of $G^*$; conversely, if $e$ is a loop of $G$, the edge $e^*$ is a cut edge of $G^*$. The graph $G^*$ is called the dual of $G$.

[10.3] P359 EULER'S FORMULA:

For a connected plane graph $G$.

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

Proof by induction

[10.3] P359 Corollary 10.20:

All planar embeddings of a connected planar graph have the same number of faces.

[10.3] P359 Corollary 10.21:

Let $G$ be a simple planar graph on at least three vertices. Then $m \leq 3n-6$. Furthermore, $m=3n-6$ if and only if every planar embedding of $G$ is a triangulation.

[10.3] P359 Corollary 10.22:

Every simple planar graph has a vertex of degress at most five.

[10.3] P359 Corollary 10.23:

$K_5$ is nonplanar

proof: If $K_5$ were planar, corollary 10.21 would give $10 = e(K_5) \leq 3v(K_5) - 6 = 9$!

[10.3] P359 Corollary 10.24:

$K_{3,3}$ is nonplanar.

proof: If $K_{3,3}$ were planar, $K_{3,3}$ has no cycle of length $< 4$, so every face of $G$ has degree $\geq 4$. By Theorem 10.10, $4f(G) \leq \sum_{f \in F} d(f) = 2e(G) = 18$

$\therefore \underbrace{f(G)}_{\text{integer}} \leq 4$

Euler's formula implies that

$$\underset{6}{v(G)} - \underset{9}{e(G)} + f(G) = 2$$

$6 - 9 + f(G) = 2 \Rightarrow f(G) = 5$

[10.2] P253 Theorem 10.10: If $G$ is a plane graph

$$\sum_{f \in F} d(f) = 2m$$

[10.5] P368 Theorem 10.30: KURATOWSKI'S THEOREM

A graph is planar iff it contains no subdivision of either $K_5$ or $K_{3,3}$.

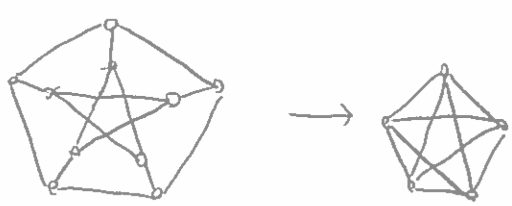
A subdivision of $K_5$ or $K_{3,3}$ is consequently called a Kuratowski subdivision.

[10.5] P368 minor: A minor of a graph is any graph obtainable from $G$ by means of a sequence of vertex and edge deletions and edge contractions?

Alternatively, consider a partition $(V_0, V_1, \ldots, V_k)$ of $V$ such that $G[V_i]$ is connected, $1 \leq i \leq k$, and let $H$ be the graph obtained from $G$ by deleting $V_0$ and shrinking each induced subgraph $G[V_i]$, $1 \leq i \leq k$, to a single vertex. The any spanning? subgraph $F$ of $H$ is a minor of $G$.

e.g. $K_5$ is a minor of the Petersen graph because it can be obtained by contracting the five 'spoke' edges of the latter graph.

Petersen graph



important:

1) any graph which contains an $F$-subdivision also has an $F$-minor

2) provided that $F$ is a graph of maximum degree three or less, any graph which has an $F$-minor also contains an $F$-subdivision.

If $F$ is a minor of $G$, we write $F \preceq G$.

Exe. find a $K_{3,3}$ minor from Petersen graph.

[10.5] P369 WAGNER'S THEOREM

A graph is planar iff it has no Kuratowski minor.

[1.1] $P_3$ loop: an edge with identical ends is called a loop
link: an edge with distinct ends is called a link
parallel edges: two or more links with the same pair of ends are said to be parallel edges
simple: a graph is simple if it has no loops or parallel edges.

[D. Battista et al.]

{1} $P_6$ drawing: a drawing $\Gamma$ of a graph (digraph) $G$ is a function which maps each vertex $v$ to a distinct point $\Gamma(v)$ and each edge $(u,v)$ to a simple open Jordan curve $\Gamma(u,v)$, with endpoints $\Gamma(u)$ and $\Gamma(v)$.

{1} $P_7$ drawing planar: a drawing $\Gamma$ is planar if no two distinct edges intersect.
A graph is planar if it admits a planar drawing

{1} $P_8$ connected: a graph is connected if there is a path between $u$ and $v$ for each pair $(u,v)$ of vertices.

cut vertex: A cut vertex in graph $G$ is a vertex whose removal disconnects $G$.
biconnected: A connected graph with no cutvertices is biconnected.
blocks: The maximal biconnected subgraphs of a graph are its blocks. (sometimes called biconnected components).
A graph is planar iff its blocks are planar.

{1} $P_9$ important: ?1) The skeleton of a convex polyhedron is a planar triconnected graph.
?2) A planar triconnected graph has a unique embedding, up to a reversal of the circular ordering of the neighbors of each vertex

{2.1} $P_{12}$ drawing convention: A drawing convention is a basic rule that the drawing must satisfy to be admissible.

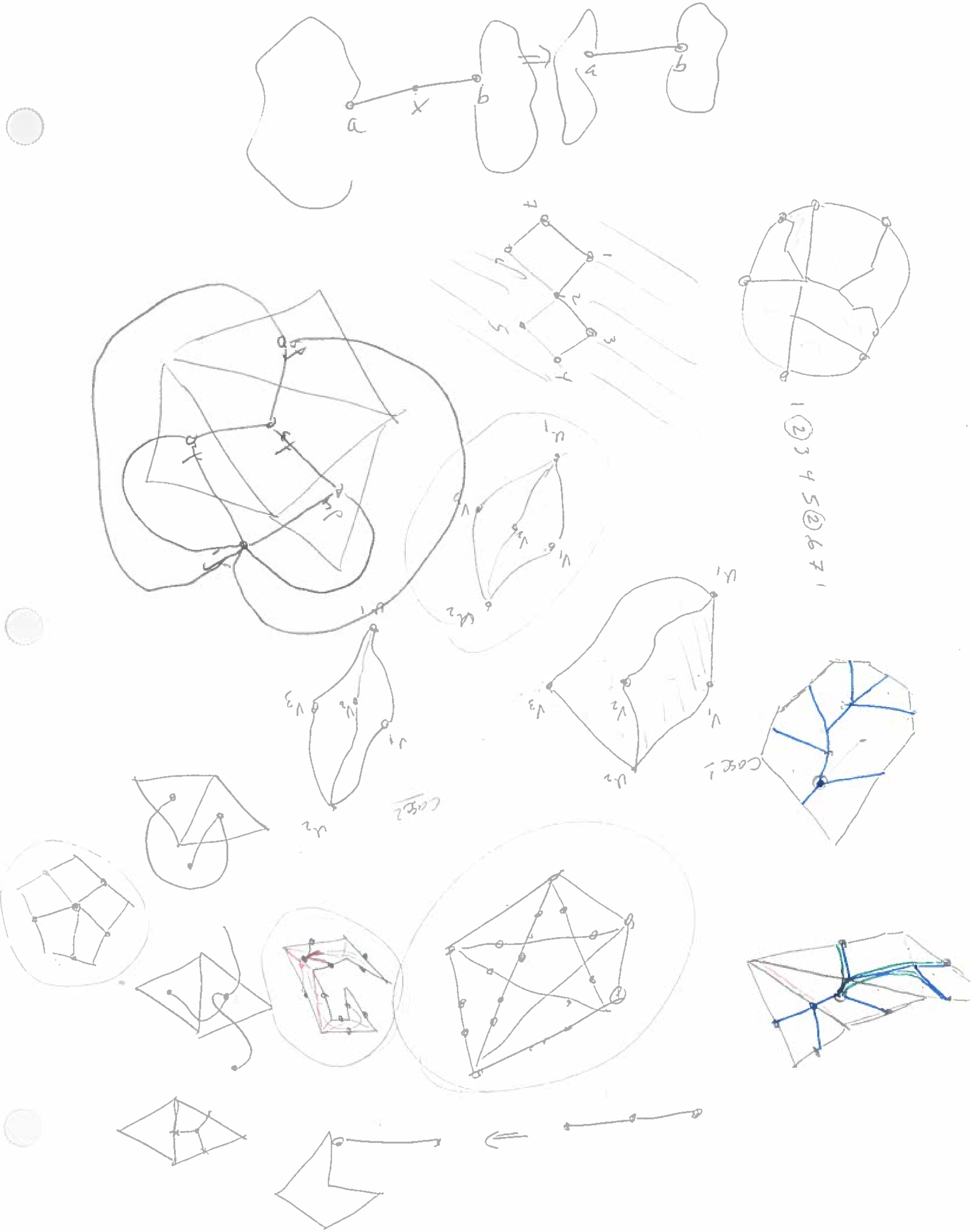
- Polyline drawing
- Straight-line drawing
- Orthogonal drawing
- Grid Drawing
- Planar Drawing
- Upward (resp. downward) Drawing

{2.1} $P_{14}$ aesthetics: aesthetics specify graph properties of the drawing that we would like to apply as much as possible, to achieve readability.

- crossings: minimize
- area: minimize (grid drawing. straight-line drawing where distance$(u,v) \geq 1$)
  convex hull
- total edge length: minimize
- maximum edge length: minimize.
- uniform edge length minimize variance of lengths of the edges
- total bends: minimize. (important for orthogonal drawings)
- maximum bends: minimize
- uniform bends: minimize
- angular resolution: maximize (straight-line drawing)
- aspect ratio: minimize longest side : shortest side
- symmetry

constraints: refer to specific subgraphs or sub drawings

efficiency: interactive applications


a
x
b
a
b

[The Left-Right Planarity Test] by Ulrik Brandes

- Efficient Planarity Testing by John Hopcroft & Robert Tarjan (1974)
- A Depth-first-search Characterization of Planarity by H. De Fraysseix & P. Rosenstiehl (1982)
- On the Realization of Complexes in Euclidean Spaces 吴文俊 Wen-Tsun Wu (1955)

{P4} There are only two significant ways to draw a simple cycle planarly, namely clockwise and counterclockwise.

{P4} Testing planarity amounts to deciding whether there is a consistent simultaneous orientation of all cycles.

{P5} In any planar drawing the back edges can be partitioned into left and right depending on whether their fundamental cycle is counterclockwise or clockwise.

{P6} In the oriented graph, we denote by $E^+(v) = \{(v,w) \in E : w \in V\}$ the set of all outgoing edges of $v \in V$, so that $E = \bigcup_{v \in V} E^+(v)$.

{P6} A DFS traversal yields a bipartition $E = T \uplus B$ of the edges, where those in $T$ are called <u>tree edges</u>, and the non-tree edges in $B$ are called <u>back edges</u>. We write $u \rightarrow v$ for $(u,v) \in T$ and $v \hookrightarrow w$ for $(v,w) \in B$.

{P6} <u>fundamental cycle</u>: $C(v \hookrightarrow w) = w \xrightarrow{+} v \hookrightarrow w$

<u>overlapping</u>. Two cycles are called overlapping, if they share an edge.

{P6} <u>Lemma 3</u> Let $G = (V, T \uplus B)$ be a DFS-oriented graph.

(1) The fundamental cycles are exactly the simple directed cycles of $G$.

(2) Two distinct fundamental cycles are either disjoint, or their intersection forms a tree path

{P7} <u>fork</u>: For two overlapping cycles, the last edge $u \rightarrow v$ on the shared tree path together with succeeding edges $e_1 = (v, w_1)$, $e_2 = (v, w_2)$ on each cycle is called their <u>fork</u>, and $v$ its <u>branching point</u>

{P7} a linearization of the cyclic ordering: It is defined by splitting the clockwise order restricted to outgoing edges at the incoming tree edge, or between any two consecutive outgoing edges if $v$ is the root of a DFS tree.

{P7} <u>nested</u>: Two overlapping fundamental cycles are called <u>nested</u>, if the part of one cycle that is not common to both is drawn completely inside the other cycle.

{P8} <u>Observation 1</u>: In a planar drawing of a DFS-oriented graph $G = (V, T \uplus B)$, two overlapping cycles are nested, if and only if they are oriented alike.

{P6} <u>return points</u>: The return points of a tree edge $v \to w \in T$ are the ancestors $u$ of $v$ with $u \xrightarrow{+} v \to w \xrightarrow{*} x \hookrightarrow u$ for some descendant $x$ of $w$.
The return points of a vertex $v \in V$ are formed by the union of all return points of outgoing edges $(v,w) \in E^+(v) \subseteq T \uplus B$.

{P8} <u>lowpoint</u>: The lowpoint of an edge is its lowest return point, if any, or its source if none exists.

{P9} <u>Observation 2</u>: In a planar drawing of a connected DFS-oriented graph $G = (V, T \uplus B)$ with the root of the DFS tree on the outer face, overlapping fundamental cycles are nested according to their lowpoint order.

{P9} <u>left, right</u>: the side of a back edge in a planar drawing is right, if its fundamental cycle is oriented clockwise, and left otherwise.

{P9} <u>LR partition</u>: Let $G = (V, T \uplus B)$ be a DFS-oriented graph. A partition $B = L \uplus R$ of its back edges into two classes, referred to as left and right, is called left-right partition, or LR partition for short, if every fork consisting of $u \to v \in T$ and $e_1, e_2 \in E^+(v)$

(1) all return edges of $e_1$ ending strictly higher than $lowpt(e_2)$ belong to one class and
(2) all return edges of $e_2$ ending strictly higher than $lowpt(e_1)$ to the other.

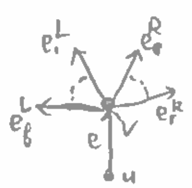
{P10} <u>Left-Right Planarity Criterion</u>: A graph is planar if and only if it admits an LR partition.

{P10} <u>aligned</u>: An LR partition is called aligned, if all return edges of a tree edge $e$ that return to $lowpt(e)$ are on the same side.

{P11} <u>Lemma 6</u>: Any LR partition can be turned into an aligned LR partition.

{P12} <u>$e_1 \prec e_2$</u>: we have to define $e_1 \prec e_2$ if and only if the lowpoint of $e_1$ is strictly lower than that of $e_2$. If both have the same lowpoint, but say, only $e_2$ has another return point, we say that $e_2$ is <u>chordal</u> and let $e_1 \prec e_2$.

{P12} <u>Definition 7 LR Ordering</u>: Given an LR partition, let $e_1^L \prec \cdots \prec e_\ell^L$ be the left outgoing edges of a vertex $v$ and $e_1^R \prec \cdots \prec e_r^R$ its right outgoing edges. If $v$ is not the root, let $u$ be its parent.
The clockwise leftright ordering, or LR ordering for short, of the edges around $v$ is defined as follows:




$(u,v)$,
$L(e_\ell^L), e_\ell^L, R(e_\ell^L), \ldots, L(e_1^L), e_1^L, R(e_1^L)$,
$L(e_1^R), e_1^R, R(e_1^R), \ldots, L(e_r^R), e_r^R, R(e_r^R)$


where $(u,v)$ is absent if $v$ is the root, and $L(e)$ and $R(e)$ denote the left and right incoming back edges whose cycles share $e$. For two back edges $b_1 = x_1 \hookrightarrow v$, $b_2 = x_2 \hookrightarrow v \in R(e)$ let $z \to x, (x,y_1), (x,y_2)$ be the fork of $C(b_1), C(b_2)$. Then, $b_1$ comes after $b_2$ in $R(e)$ if and only if $(x,y_1) \prec (x,y_2)$. If $b_1, b_2 \in L(e)$, the order is reversed.

{$P_{12}$} Lemma 8. Given an LR partition, LR ordering yields a planar embedding.

{$P_{14}$} Corollary 9. Let $G = (V, T \uplus B)$ be a DFS-oriented graph. For a pair of back edges $b_1, b_2 \in B$ with overlapping fundamental cycles let $v_1 \to \cdots \to v_k$ be the tree path of the intersection and $(v_{k-1}, v_k), e_1, e_2$ the corresponding fork with $e_1 \xrightarrow{*} b_1$ and $e_2 \xrightarrow{*} b_2$. Then, $b_1$ and $b_2$ are subject to

- a different-constraint, iff lowpoint$(e_2)$ < lowpoint$(b_1)$ and lowpoint$(e_1)$ < lowpoint$(b_2)$
- a same-constraint, iff lowpoint$(e') < \min\{$lowpoint$(b_1)$, lowpoint$(b_2)\}$ for some $e' = (v_i, w) \in T \uplus B$, $1 < i < k$, $w = v_{i+1}$


(a) different-constraint

{$P_{16}$} Definition 10. Let $G = (V, T \uplus B)$ be a DFS-oriented graph such that each pair of back edges $b_1, b_2 \in B$ is subject to at most one type of constraint. The signed graph $C(G) = (B, E(C); \sigma : E(C) \to \{-1, +1\})$ with

$$\sigma(b_1, b_2) = \begin{cases} -1 & \text{if } b_1, b_2 \in B \text{ are subject to a different constraint} \\ +1 & \text{if } b_1, b_2 \in B \text{ are subject to a same constraint} \end{cases}$$

is called constraint graph of $G$.

{$P_{16}$} If any pair of back edges is subject to both a same-constraint and a different-constraint, no LR partition can exist and hence the graph is non-planar.

{$P_{18}$} Algorithm 1: Left-Right Planarity Algorithm

```
input: simple, undirected graph G = (V, E)
output: planar embedding (halts if graph is not planar)
if |V| > 2 and |E| > 3|V| - 6 then HALT: not planar

▼ orientation
  for s ∈ V do
    if height[s] = ∞ then
      height[s] ← 0; append Roots ← s
      DFS1(s)

▼ testing
  sort adjacency lists according to non-descending nesting_depth
  for s ∈ Roots do DFS2(s)

▼ embedding
  for e ∈ E do nesting_depth[e] = sign(e) · nesting_depth[e]
  sort adjacency lists according to non-decreasing nesting_depth
  for s ∈ Roots do DFS3(s)

where
integer sign(edge e)
  if ref[e] ≠ ⊥ then
    side[e] ← side[e] · sign(ref[e])
    ref[e] ← ⊥
```

# HOW TO DRAW A PLANAR GRAPH ON A GRID

by H. DE FRAYSSEIX, J. PACH and R. POLLACK.

**Theorem 1**

The paper shows that every plane graph with $n$ vertices has a Fáry embedding (i.e., straight-line embedding) on the $2n-4$ by $n-2$ grid and provides an $O(n)$ space, $O(n \log n)$ time algorithm to effect this embedding.

It also shows that any set $F$, which can support a Fáry embedding of every planar graph of size $n$, has cardinality at least $n + (1 - o(1))\sqrt{n}$.

① Run Hopcroft–Tarjan planarity testing algorithm
outputs a topological embedding of a planar graph.

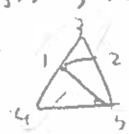
② maximal plane graph – triangulated (all faces are triangles).

**Proposition 1**

Given a maximal planar graph $G$ and a face $uvw$, there is a labelling of the vertices, $v_1 = u$, $v_2 = v$, $v_3, \dots, v_n = w$ and a Fáry embedding such that the convex hull of $\{f(v_1), f(v_2), f(v_3), \dots, f(v_k)\}$ is the same as the convex hull of $\{f(v_1), f(v_2), f(v_k)\}$ for $k = 4, \dots, n$

**Proposition 2**

Given a maximal planar graph $G$ and a face $uvw$, there is a labelling of the vertices, $v_1 = u$, $v_2 = v$, $v_3, \dots, v_n = w$ and a Fáry embedding $f$ such that the convex hull of $\{f(v_1), f(v_2), f(v_3), \dots, f(v_k)\}$ is the same as the convex hull of $\{f(v_{k-2}), f(v_{k-1}), f(v_k)\}$, for $k = 4, \dots, n$.




**Proposition 3**

Given a maximal planar graph $G$ and a face $uvw$, there is a labelling of the vertices $v_1 = u$, $v_2 = v$, $v_3, \dots, v_n = w$ and a Fáry embedding $f$ such that the boundary of the convex hull of $\{f(v_1), f(v_2), f(v_3), \dots, f(v_k)\}$ is a cycle in $G$ and $f(v_{k+1})$ is not contained in the convex hull of $\{f(v_1), f(v_2), f(v_3), \dots, f(v_k)\}$.

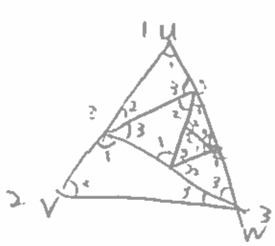
**Proposition 4** Schnyder

Given a maximal planar graph $G$ with exterior face $uvw$, there is a labelling of the angles of the internal triangles with labels 1, 2 and 3 such that

(i) each triangle has labels 1, 2 and 3 in counterclockwise order.

(ii) all angles at $u$, $v$ and $w$ are labelled 1, 2 and 3, respectively.

(iii) around each internal vertex the angles of each label appear in a single block.




?

**Theorem 2** If $F$ is universal for planar graphs with $n$ vertices then

$$|F| > n + (1 - o(1))\sqrt{n}$$

**outerplanar** A planar graph which can be obtained from a simple cycle by adding some of its internal diagonals is called outerplanar.

**Proposition 5** Every set of $n$ points in the plane, in general position, supports every outerplanar graph with $n$ vertices. Moreover, this property characterizes the outerplanar graphs.

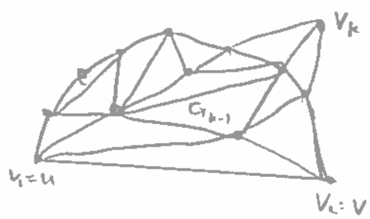
**Lemma** Let $G$ be a simple planar graph embedded in the plane and $u = u_1, u_2, \dots, u_k = v$ be a cycle of $G$. Then there exists a vertex $w'$ (resp. $w''$) on the cycle, different from $u$ and $v$ and not adjacent to any inside chord (resp. outside chord).

**Canonical representation lemma for plane graphs**

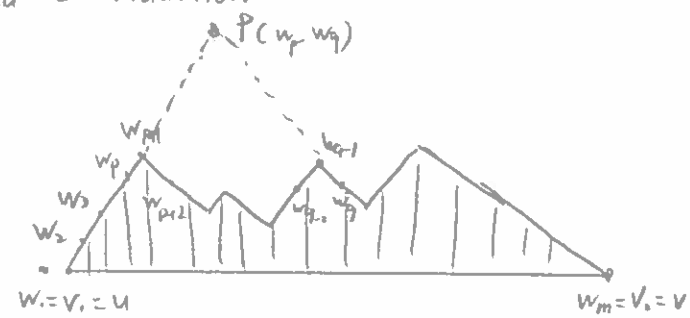
Let $G$ be a maximal planar graph embedded in the plane with exterior face $u, v, w$. Then there exists a labelling of the vertices $v_1 = u, v_2 = v, v_3, \dots, v_n = w$ meeting the following requirements for every $4 \leq k \leq n$.

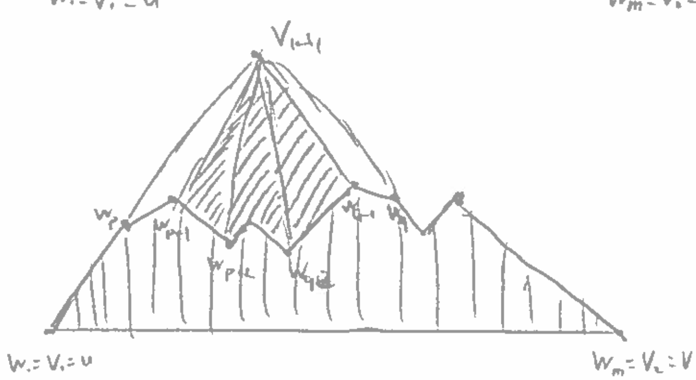
(i) The subgraph $G_{k-1} \subseteq G$ induced by $v_1, v_2, \dots, v_{k-1}$ is 2-connected, and the boundary of its exterior face is a cycle $C_{k-1}$ containing the edge $uv$;

(ii) $v_k$ is in the exterior face of $G_{k-1}$, and its neighbours in $G_{k-1}$ form an (at-least 2-element) subinterval of the path $C_{k-1} - uv$.




Key idea of induction




To realize this goal, assume that for each vertex $w_i$ on the exterior face of $G_k$ we have already defined a subset $M(k, w_i) \subseteq V(G_k)$ such that

(a) $w_j \in M(k, w_i)$ iff $j \geq i$

(b) $M(k, w_1) \supset M(k, w_2) \supset \dots \supset M(k, w_m)$

(c) For any nonnegative numbers $\alpha_1, \alpha_2, \dots, \alpha_m$, if we sequentially translate all vertices in $M(k, w_i)$ with distance $\alpha_i$ to the right ($i = 1, 2, \dots, m$), then the embedding of $G_k$ remains a Fáry embedding.

# Embedding Planar Graphs on the Grid by Walter Schnyder

This paper shows that each plane graph of order $n \geq 3$ has a straight line embedding on the $n-2$ by $n-2$ grid. This embedding is computable in time $O(n)$.

cf. I. Fáry, On straight line representation of planar graphs, Acta Sci. Math. Szeged 11 (1948), 229-233.

Characterization: planar graphs are graphs whose incidence relation is the intersection of three total orders.

<u>THEOREM 1.1</u> Let $\lambda_1, \lambda_2, \lambda_3$ be three pairwise non parallel straight lines in the plane. Then each plane graph has a straight line embedding in which any two disjoint edges are separated by a straight line parallel to $\lambda_1$, $\lambda_2$, or $\lambda_3$.

<u>THEOREM 1.2</u> Each plane graph with $n \geq 3$ vertices has a straight-line embedding on the $n-2$ by $n-2$ grid.

## Barycentric representations

A barycentric representation of a graph $G$ is an injective function $v \in V(G) \longrightarrow (v_1, v_2, v_3) \in \mathbb{R}^3$ that satisfies the conditions:

(1) $v_1 + v_2 + v_3 = 1$ for all vertices $v$

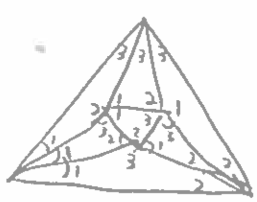
(2) For each edge $\{x, y\}$ and each vertex $z \notin \{x, y\}$, there is some $k \in \{1, 2, 3\}$ such that $x_k < z_k$ and $y_k < z_k$

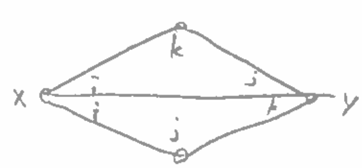
## normal labeling

A normal labeling of a triangular graph $G$ is a labeling of the angles of $G$ with the labels 1, 2, 3 satisfying the conditions

(1) Each elementary triangle of $G$ has an angle labeled 1, an angle labeled 2, and an angle labeled 3. The corresponding vertices appear in counterclockwise order.

(2) The labels of the angles of each interior vertex $v$ of $G$ form, in counterclockwise order, a nonempty interval of 1's followed by a nonempty interval of 2's by a nonempty interval of 3's.



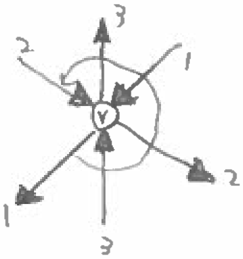



THEOREM 4.2. Each triangular graph has a normal labeling.

realizer  A realizer of a triangular graph $G$ is a partition of the interior edges of $G$ in three sets $T_1, T_2, T_3$ of directed edges such that for each interior vertex $v$ there holds:

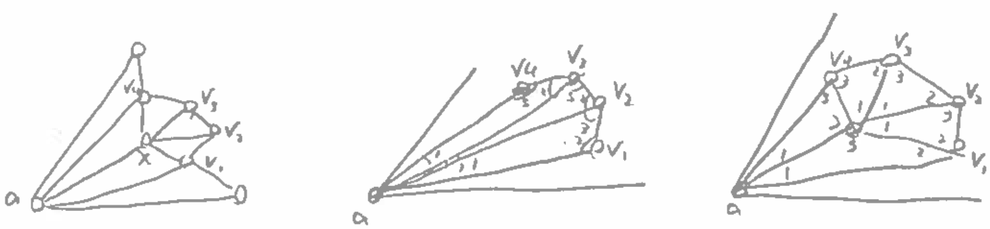
(1) $v$ has outdegree one in each of $T_1, T_2, T_3$

(2) The counterclockwise order of the edges incident on $v$ is: leaving $T_1$, entering $T_3$, leaving in $T_2$, entering in $T_1$, leaving in $T_3$, entering in $T_2$.




The edge $\{x,y\}$ is contractible if $x$ and $y$ have exactly two common neigbours

Prove THEOREM 4.2 by induction.




THEOREM 4.5

Let $G$ be a triangular graph with at least four vertices and let $T_1, T_2, T_3$ be a realizer of $G$. Then each $T_i$ is a tree including all interior vertices and exactly one exterior vertex and all edges of $T_i$ are directed toward this exterior vertex The exterior vertices belonging to $T_1, T_2, T_3$ are distinct and appear in counterclockwise order.

THEOREM 4.6

If $T_1, T_2, T_3$ is a realizer of a triangular graph, then for $i=1,2,3$ the relation $T_i \cup T_{i+1}^{-1} \cup T_{i+2}^{-1}$ has no directed cycle (indices are modulo 3).

THEOREM 6.1

The function $f: v \in V(G) \longrightarrow \frac{1}{2n-5}(v_1, v_2, v_3)$ is a barycentric representation of $G$ and the labeling of $G$ that is induced by $f$ is identical to the given labeling of $G$.

COROLLARY 6.2

Let $a, b$ and $c$ denote the roots of $T_1, T_2, T_3$. Then for any choice of noncolinear positions of $a, b$ and $c$ the mapping

$$f: v \longrightarrow \frac{1}{2n-5}(v_1 a + v_2 b + v_3 c)$$

is a straight line embedding of $G$ in the plane spanned by $a, b, c$.

PROPOSITION 6.3

The mapping $v \in V(G) \longrightarrow (v_1, v_2)$ is a straight line embedding of $G$ on the $2n-5$ by $2n-5$ grid.

HOW TO DRAW A GRAPH By W.T. TUTTE

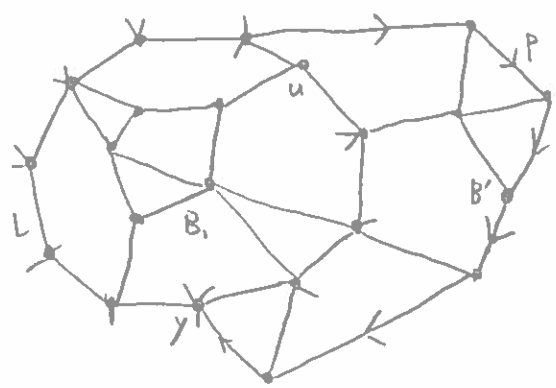
<u>3-connected</u>: A graph $G$ is 3-connected (nodally 3-connected) if it is simple and non-separable and satisfies the following condition; if $G$ is the union of two proper subgraphs $H$ and $K$ such that $H \cap K$ consists solely of two vertices $u$ and $v$, then one of $H$ and $K$ is a link-graph (arc-graph) with ends $u$ and $v$.

<u>peripheral polygon</u>: Let $J$ be a polygon of $G$ and let $\beta(J)$ denote the number of bridges of $J$ in $G$. If $\beta(J) \leq 1$ we call $J$ a peripheral polygon of $G$.

(2.1) Let $G$ be a nodally 3-connected graph. Let $J$ be a polygon of $G$ and $B$ any bridge of $J$ in $G$. Then either $J$ is a peripheral or $J$ has another bridge $B'$ which does not avoid $B$.

<u>avoid</u>: If one of residual arc-graphs of $B$ in $J$ includes all the vertices of attachment of a second bridge $B'$ of $J$ in $G$ we say that $B'$ avoids $B$. Then $B$ avoids $B'$.

(2.2) Let $G$ be a nodally 3-connected graph. Let $K_1$ be a polygon of $G$, $B_1$ a bridge of $K_1$ in $G$, $C$ a subgraph of $B_1$, and $L$ a branch of $G$ in $K_1$. Then we can find a peripheral polygon $J$ of $G$ such that $L \subset J$ and $J \cap C \subseteq K_1 \cap C$.




(2.3) Let $G$ be a nodally 3-connected graph which is not a polygon or a link-graph, and let $L$ be a branch of $G$. Then we can find two peripheral polygons $J_1$ and $J_2$ of $G$ such that $J_1 \cap J_2 = L$.

(2.4) Let $G$ be a nodally 3-connected graph, $K$ a polygon of $G$, $B$ a bridge of $K$ in $G$, and $L$ a branch of $G$ contained in $K$. Let $J_1$ and $J_2$ be peripheral polygons of $G$ such that $L \subseteq J_1 \cap J_2$ and neither $B \cap J_1$ nor $B \cap J_2$ is a subgraph of $K$. Then we can find a peripheral polygon $J_3$, distinct from $J_1$ and $J_2$, such that $L \subset J_3$.

(2.5) Let G be a nodally 3-connected non-null graph. Then we can find a set of $p_1(G)$ independent peripheral cycles of G.

peripheral cycle : Consider the set of cycles of a connected graph G. The rank of this set, the maximum number of cycles independent under mod-2 addition, is

$$p_1(G) = \alpha_1(G) - \alpha_0(G) + 1$$

We refer to the elementary cycles associated with a peripheral polygon as a peripheral cycle.

(2.6) Let G be a nodally 3-connected non-null graph, with at least two edges, which is not a polygon. Suppose that no edge of G belongs to more than two distinct peripheral polygons. Then G has just $p_1(G) + 1$ distinct peripheral cycles, and they constitute a planar mesh of G.

(2.7) A peripheral polygon K of a non-seperable graph G belongs to every planar mesh of G.

(2.8) If M is a planar mesh of a nodally 3-connected graph G, then each member of M is peripheral.

(2.7) + (2.8) show that a nodally 3-connected graph has at most one planar mesh.

representation : We call H a representation of G in $\Pi$ (closed plane) if it satifies the following conditions

(i) No edge of H contains any vertex of H.

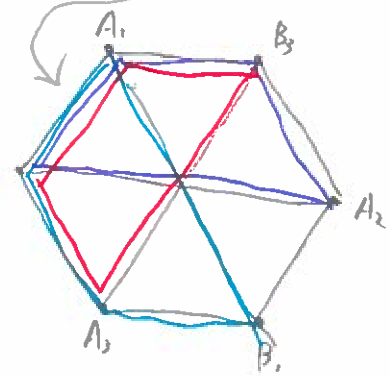
(ii) If e and e' are distinct edges of G, then f(e) and f(e') are disjoint.

A graph G is said to be planar if it has a representation in $\Pi$

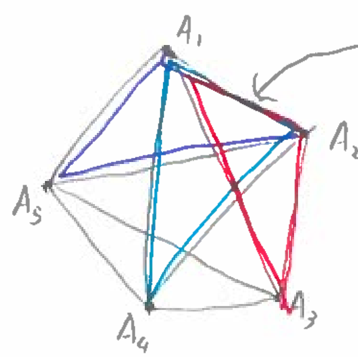
(3.1) Each peripheral polygon of H bounds a face of H.

(3.2) If a graph G has three distinct peripheral polygons with a common edge, then G is non-planar.

Kuratowski graph of Type I.




Kuratowski graph of Type II

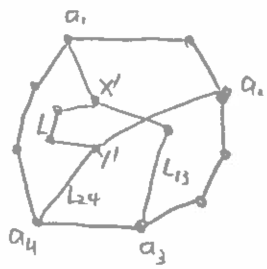



(4.1) Every Kuratowski graph is non-planar.

COROLLARY. Any graph having a Kuratowski subgraph is non-planar.

<u>Crossing diagonals:</u> Let $J$ be a polygon of a graph $G$. Let $a_1, a_2, a_3, a_4$ be distinct vertices of $J$ such that $a_1$ and $a_3$ separate $a_2$ from $a_4$ on $J$. Let $L_{13}$ and $L_{24}$ be disjoint arc-graphs of $G$ spanning $J$. Then we say that $L_{13}$ and $L_{24}$ are crossing diagonals of $J$.

(5.1) Given a peripheral polygon of $G$ with a pair of crossing diagonals we can find a Kuratowski subgraph of $G$ of Type I.




($a_1, a_2, a_3, a_4, x', y'$ form a Kuratowski graph of Type I)

(5.2) Let $J$ be a peripheral polygon of a graph $G$. Let $a$, $b$, and $c$ be distinct vertices of $J$. Let $Y_1$, $Y_2$ be $Y$-graphs of $G$, each with ends $a$, $b$, and $c$, which spans $J$. Suppose further that $Y_1 \cap Y_2$ consists solely of the vertices $a$, $b$, and $c$. Then we can find a Kuratowski subgraph of $G$.

## Barycentric mappings

If $n < i \leq m$ let $A(i)$ be the set of all vertices of $G$ adjacent to $v_i$, that is joined to $v_i$ by an edge. For each $v_j$ in $A(i)$ let a unit mass $m_j$ be placed at point $f(v_j)$. Then $f(v_i)$ is required to be the centroid of the masses $m_j$.

Denoting the coordinates of $f(v_i)$, $1 \leq i \leq m$, by $(x_i, y_i)$.
Define a matrix $K(G) = \{C_{ij}\}$, $1 \leq (i, j) \leq m$, as follows.
If $i \neq j$ then $C_{ij}$ is minus the number of edges joining $v_i$ and $v_j$.
If $i = j$ then $C_{ij}$ is the degree of $v_i$.

Then the foregoing requirement specifies the coordinates $x_j$ and $y_j$, for $n < j \leq m$, as the solutions of the equations

$$(5) \qquad \sum_{j=1}^{m} C_{ij} x_j = 0$$

$$(6) \qquad \sum_{j=1}^{m} C_{ij} y_j = 0.$$

where $n < i < m$.

$\varphi(i)$: Choose a line $L$ in the plane and define $\varphi(i)$, $1 \le i \le m$, as the perpendicular distance of $f(v_i)$ from $L$, counted positive on one side of $L$ and negative on the other.

$\varphi$-active: We call $v_i$ $\varphi$-active if there is an adjacent vertex $v_j$ of $G$ such that $\varphi(j) \neq \varphi(i)$

positive $\varphi$-poles: The nodes $v_i$ of $J$ with the greatest value of $\varphi(i)$ are the positive $\varphi$-poles of $G$. The number of positive $\varphi$-poles is either 1 or 2.

rising (falling) $\varphi$-path: Let $P$ be a simple path in $G$. We call $P$ a rising (falling) $\varphi$-path if each vertex of $P$ other than the last corresponds to a smaller (greater) value of the function $\varphi(i)$ than does the immediately succeeding vertex.

(6.1) Suppose that $v_i$, where $n < i \le m$, is a $\varphi$-active vertex. Then it has adjacent vertices $v_j$ and $v_k$ such that $\varphi(v_j) < \varphi(v_i) < \varphi(v_k)$.

(6.2) Let $v_i$ be a $\varphi$-active vertex. Then we can find a rising $\varphi$-path $P$ from $v_i$ to a positive $\varphi$-pole, and a falling $\varphi$-path $P'$ from $v_i$ to a negative $\varphi$-pole.

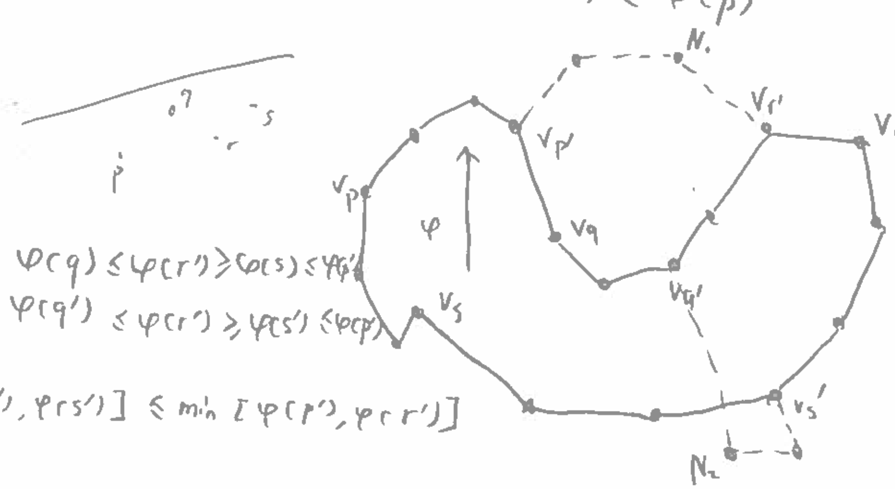
(7.1) Every node of $G$ is $\varphi$-active.

(7.2) Suppose $v_i \notin V(J)$. Then $f(v_i)$ is in the interior of $Q$.

(8.1) Let $K$ be a peripheral polygon of $G$ such that $V(K)$ includes just three nodes $x$, $y$, and $z$ of $G$. Then $f(x)$, $f(y)$, and $f(z)$ are not collinear.

(8.2) Let $K$ be a peripheral polygon of $G$. Let $v_p$, $v_q$, $v_r$, and $v_s$ be nodes of $G$ in $V(K)$ such that $v_p$ and $v_r$ separate $v_q$ and $v_s$ in $K$. Then it is not true that

$$\varphi(p) \ge \varphi(q) \le \varphi(r) \ge \varphi(s) \le \varphi(p) \qquad (7)$$




✓ (8) $\varphi(p') \ge \varphi(q) \le \varphi(r') \ge \varphi(s) \le \varphi(p')$

✓ (9) $\varphi(p') \ge \varphi(q') \le \varphi(r') \ge \varphi(s') \le \varphi(p')$

$\varphi(j) < \max[\varphi(q'), \varphi(s')] \le \min[\varphi(p'), \varphi(r')]$

(8.3) The nodes of any peripheral polygon $K$ of $G$ are mapped by $f$ onto distinct points of the plane, no three of which are collinear.

(8.4) Let $L$ be a branch of $G$ having just $t \geq 1$ internal vertices, and let its ends be $a$ and $b$. Then $f(a)$ and $f(b)$ are distinct and $f$ maps the internal vertices onto $t$ distinct points of the segment $f(a)f(b)$ subdividing it into $t+1$ equal parts. Moreover, the order of the vertices from $a$ to $b$ in $L$ agrees with that of their images in $f(a)f(b)$

(8.5) Let $K$ be any peripheral polygon of $G$. Then $f$ maps the nodes of $G$ in $K$ onto the vertices of a (geometrical) convex polygon $Q_K$ so that the cyclic order of nodes in $K$ agrees with that of vertices in $Q_K$.

(8.6) Let $e$ be any edge of $R$. Then just two distinct peripheral polygons of $G$ pass through $e$, and the two corresponding regions $R_K$ lie on opposite sides of the segment $f(e)$.

**Barycentric representations.**

(9.1) $\underline{\delta(A)} = 1$ for each $A$ in $\underline{S}$. → $|H|$'s complentary set in the plane.

point

number of distinct peripheral polygons $K$ of $G$ such that $A \in R_K$.

(9.2) Let $G$ be a nodally 3-connected graph having no Kuratowski subgraph. Let $J$ be a peripheral polygon of $G$ which includes just $n \geq 3$ nodes of $G$. Let $Q$ be an $n$-sided convex polygon in the Euclidean plane. Then there is a unique barycentric representation of $G$ on $Q$ mapping the nodes of $G$ occuring on $J$ onto the vertices of $Q$ in any orbitrary specified way preserving the cyclic order.

(9.3) Let $G$ be a nodally 3-connected graph having at least one polygon. Then if $G$ has no Kuratowski subgraph we can construct a convex representation of $G$.

**Straight representations**

(10.1) Let $G$ be the union of two proper subgraphs $H$ and $K$ such that $H \cap K$ is either null or a vertex-graph. Let $M_H$ and $M_K$ be planar meshes of $H$ and $K$ respectively. Then $M_H \cup M_K$ is a planar mesh of $G$. Moreover, any planar mesh of $G$ can be represented in this form.

(10.6) Let $G$ be a graph having a planar mesh $M$. Then each subgraph of $G$ has a planar mesh.

(10.7) If a graph has a planar mesh it has no Kuratowski subgraph.

(10.8) Let $G$ be any simple graph having a planar mesh. Then by adding new links to $G$, with ends in $V(G)$, we can construct a nodally 3-connected graph $T$ having a planar mesh.

(10.9) If $G$ is a simple graph having a planar mesh we can find a straight representation of $G$ in this plan

{MIT 6.889} by Erik Demaine, Shay Mozes, Christian Sommer, Siamak Tazari

"Solve your favourite problems faster for graphs that matter!"

<u>Survey</u>:

| Problems | <u>general</u> | vs. <u>planar</u> |
| --- | --- | --- |
| – single-source shortest paths (arbitrary weights) | $O(nm)$ [Bellman-Ford] | $O\left(n \frac{\lg^2 n}{\lg \lg n}\right)$ [Mozes & Wulff-Nilsen – ESA 2010] |
| – nonnegative weights | $O(n \lg n + m)$ [Dijkstra] + [Fredman & Tarjan] – JACM 1987 | $O(n)$ [Henzinger, Klein, Rao, Subramanian – JCSS 1997] |
| – maximum flow | $O(nm \lg n)$ [Goldberg & Tarjan 1986] $O(m^{3/2} \lg n \lg u)$ [Goldberg & Rao 1997] | $O(n \lg n)$ [Borradaile & Klein – JACM 2009] |
| – undirected | | $O(n \lg \lg n)$ [Italiano, Nussbaum, Sankowski, Wulff-Nilsen – STOC 2011] |
| – multiterminal | | $O(n \lg^3 n)$ [Borradaile, Klein, Mozes, Nussbaum, Wulff-Nilsen – FOCS 2011] |
| – min spanning tree | $O(n)$ rand. [Karger, Klein, Tarjan 1985] | $O(n)$ det |

<< Algorithm Design >> by Jon Kleinberg and Éva Tardos

## § Chapter 13 : Randomized Algorithms

When one thinks about random process, it is usually in one of two distinct ways.

1) average-case analysis (randomly generated input)

2) consider algorithms that behave randomly $\longrightarrow$ randomized algorithm

### 13.1 A First Application: Contention Resolution

Suppose $n$ processes $P_1, P_2, \dots, P_n$, each competing for access to a single shared database. time as being divided into discrete rounds. Database can be accessed by at most one process in a single round. Processes can't communicate with one another at all.

Algorithm

each process will attempt to access the database in each round with probability $p$, independently of the decisions of the other processes.

Analyzing the Algorithm

let $\mathcal{A}[i,t]$ denote the event that $P_i$ attempts to access database in round $t$. $Pr[\mathcal{A}[i,t]] = p$

let $\mathcal{S}[i,t]$ denote the event that $P_i$ succeeds in accessing the database in round $t$. $Pr[\overline{\mathcal{A}[i,t]}] = 1-p$

$$Pr[\mathcal{S}[i,t]] = Pr[\mathcal{A}[i,t]] \cdot \prod_{j \neq i} Pr[\overline{\mathcal{A}[j,t]}] = p(1-p)^{n-1}$$

$$f(p) = p(1-p)^{n-1}$$

$$f'(p) = (1-p)^{n-1} - p \cdot (n-1)(1-p)^{n-2}$$

$$(1-p)^{n-1} - (n-1) \cdot p(1-p)^{n-2} \geq 0$$

$$(1-p) \geq (n-1)p$$

$$\frac{1}{p} - 1 \geq n-1$$

$$p \leq \frac{1}{n}$$

we set $p = \frac{1}{n}$. $\quad Pr[\mathcal{S}[i,t]] = \frac{1}{n}(1-\frac{1}{n})^{n-1}$

(13.1)

(a) The function $(1-\frac{1}{n})^n$ converges monotonically from $\frac{1}{4}$ up to $\frac{1}{e}$ as $n$ increases from 2.

(b) The function $(1-\frac{1}{n})^{n-1}$ converges monotonically from $\frac{1}{2}$ down to $\frac{1}{e}$ as $n$ increases from 2.

$$\therefore \quad \frac{1}{en} \leq \underbrace{Pr[\mathcal{S}[i,t]]}_{\Theta(\frac{1}{n})} \leq \frac{1}{2n}$$

let $\mathcal{F}[i,t]$ denote the "failure event" that process $P_i$ does not succeed in any of the rounds 1 through $t$.

$$Pr[\mathcal{F}[i,t]] = Pr\left[\bigcap_{r=1}^{t} \overline{\mathcal{S}[i,r]}\right] = \left[1 - \frac{1}{n}(1-\frac{1}{n})^{n-1}\right]^t \leq (1-\frac{1}{en})^t \underset{\text{set } t = \lceil en \rceil}{\leq} (1-\frac{1}{en})^{en} \leq \frac{1}{e}$$

$t = \lceil en \rceil \cdot c \ln n$

$$Pr[F[i,t]] \leq \left(1 - \frac{1}{en}\right)^t = \left(\left(1 - \frac{1}{en}\right)^{\lceil en \rceil}\right)^{c \ln n} \leq e^{-c \ln n} = n^{-c}$$

Conclusion: After $\Theta(n)$ rounds ($t = \lceil en \rceil$), the probability that $P_i$ has not succeeded is bounded by a constant ($\frac{1}{e}$); and between then and $\Theta(n \ln n)$, this probability drops to a quantity that is quite small, bounded by an inverse polynomial in $n$.

## Waiting for All Processes to Get Thougl

$$\mathcal{F}_t = \bigcup_{i=1}^{n} F[i,t]$$

(13.2) (The Union Bound) Given events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$, we have

$$Pr\left[\bigcup_{i=1}^{n} \mathcal{E}_i\right] \leq \sum_{i=1}^{n} Pr[\mathcal{E}_i].$$

$$\therefore \quad Pr[\mathcal{F}_t] \leq \sum_{i=1}^{n} Pr[F[i,t]]$$

choose $t = \lceil en \rceil (c \ln n)$, $Pr[F[i,t]] \leq n^{-c}$

choose $t = 2\lceil en \rceil \ln n$

$$Pr[\mathcal{F}_t] \leq \sum_{i=1}^{n} Pr[F[i,t]] \leq n \cdot n^{-2} = n^{-1}$$

(13.3) With probability at least $1 - n^{-1}$, all processes succeed in accessing the database at least once within $t = 2\lceil en \rceil \ln n$ rounds.

## 13.2 Finding the Global Minimum Cut

undirected graph $G = (V, E)$, define a cut of $G$ to be a partition of $V$ into two non-empty sets $A$ and $B$. For a cut $(A, B)$, the size of $(A, B)$ is the number of edges with one end in $A$ and the other in $B$. A global minimum cut is a cut of minimum size.

(13.4) There is a polynomial-time algorithm to find a global min-cut in an undirected graph $G$.

convert to directed graph

fix $s$. for every $t \in V - \{s\}$, run push-relabel.

the best among these will be a global min-cut of $G$.

David Karger 1992.

Algorithm. (Contraction Algorithm)

└ works with connected multigraph.

The Contraction Algorithm applied to a multigraph $G = (V, E)$:

For each node $v$, we will record the set $S(v)$ of the nodes that have been contracted into $v$.

Initially $S(v) = \{v\}$ for each $v$

If $G$ has two nodes $v_1$ and $v_2$, then return the cut $(S(v_1), S(v_2))$.

Else choose an edge $e = (u, v)$ of $G$ uniformly at random

Let $G'$ be the graph resulting from the contraction of $e$, with a new node $z_{uv}$ replacing $u$ and $v$.

Define $S(z_{uv}) = S(u) \cup S(v)$

Apply the Contraction Algorithm recursively to $G'$

End if

## Analyzing the Algorithm

(13.5) The Contraction Algorithm returns a global min-cut of $G$ with probability at least $\frac{1}{\binom{n}{2}}$

Suppose the global min-cut has size $k$, a set $F$ of $k$ edges with one end in $A$ and the other in $B$.

$\hookrightarrow$ every node in $G$ has degree at least $k$.

We want an upper bound on the probability that an edge in $F$ is contracted, and for this we need a lower bound on the size of $E$.

$$|E| \geq \frac{1}{2} kn$$

Hence the probability than an edge in $F$ is contracted is at most

$$\frac{k}{\frac{1}{2}kn} = \frac{2}{n}.$$

Consider the situation after $j$ iterations, there are $n-j$ supernodes in $G'$.

Thus $G'$ has at least $\frac{1}{2}k(n-j)$ edges. So the probability that an edge of $F$ is contracted in the next iteration $j+1$ is at most.

$$\frac{k}{\frac{1}{2}k(n-j)} = \frac{2}{n-j}$$

We write $\mathcal{E}_j$ for the event that an edge of $F$ is not contracted in iteration $j$, then we have shown $Pr[\mathcal{E}_1] \geq 1 - \frac{2}{n}$ and $Pr[\mathcal{E}_{j+1} \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cap \cdots \cap \mathcal{E}_j] \geq 1 - \frac{2}{n-j}$

We are interested in lower-bounding the quality $Pr[\mathcal{E}_1 \cap \mathcal{E}_2 \cdots \cap \mathcal{E}_{n-2}]$.

$$Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2 \mid \mathcal{E}_1] \cdots Pr[\mathcal{E}_{j+1} \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cdots \cap \mathcal{E}_j] \cdots Pr[\mathcal{E}_{n-2} \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cdots \cap \mathcal{E}_{n-3}]$$

$$\geq \left(1 - \frac{2}{n}\right) \cdot \left(1 - \frac{2}{n-1}\right) \cdots \left(1 - \frac{2}{n-j}\right) \cdots \left(1 - \frac{2}{3}\right)$$

$$= \left(\frac{n-2}{n}\right) \cdot \left(\frac{n-3}{n-1}\right) \left(\frac{n-4}{n-2}\right) \cdots \left(\frac{3}{5}\right)\left(\frac{2}{4}\right)\left(\frac{1}{3}\right)$$

$$= \frac{2}{n(n-1)} = \binom{n}{2}^{-1}$$

So we know that a single run of the Contraction Algorithm fails to find a global min-cut with probability at most $(1 - 1/\binom{n}{2})$.

If we run the algorithm $\binom{n}{2}$ times,

$$\left(1 - 1/\binom{n}{2}\right)^{\binom{n}{2}} \le \frac{1}{e}$$

If we run the algorithm $\binom{n}{2} \ln n$ times

$$\left[\left(1 - 1/\binom{n}{2}\right)^{\binom{n}{2}}\right]^{\ln n} \le e^{-\ln n} = n^{-1}$$

## Further Analysis: The Number of Global Minimum Cuts

Given an undirected graph $G = (V, E)$, what is the maximum number of global min-cuts it can have (as a function of $n$)?

undirected graph has $\binom{n}{2}$ global min-cuts.

(13.6) An undirected graph $G = (V, E)$ on $n$ nodes has at most $\binom{n}{2}$ global min-cuts.

Let $G$ be a graph, and let $C_1, \dots, C_r$ denote all its global min-cuts.
Let $\mathcal{E}_i$ denote the event that $C_i$ is returned by the Contraction Algorithm,
let $\mathcal{E} = \bigcup_{i=1}^{r} \mathcal{E}_i$ denote the event that the algorithm returns any global min-cut.

$$\Pr[\mathcal{E}] = \Pr[\cup_{i=1}^{r} \mathcal{E}_i] = \sum_{i=1}^{r} \Pr[\mathcal{E}_i] \ge r / \binom{n}{2}$$

$$\Pr[\mathcal{E}] \le 1 \implies r \le \binom{n}{2}$$

## 13.3 Random Variables and Their Expectations

random variable: Given a probability space, a random variable $X$ is a function from the underlying sample space to the natural numbers, such that for each natural number $j$, the set $X^{-1}(j)$ of all sample points taking value $j$ is an event.

$\Pr[X = j]$ as loose shorthand for $\Pr[X^{-1}(j)]$.

expectation: the "average value" assumed by $X$.

$$E[X] = \sum_{j=0}^{\infty} j \cdot \Pr[X = j].$$

$$S = \sum_{j=1}^{\infty} j(1-p)^j = 1\cdot(1-p)^1 + 2(1-p)^2 + 3(1-p)^3 + \cdots$$

$$(1-p)S = 1(1-p)^2 + 2(1-p)^3 + 3(1-p)^4 + \cdots$$

$$S - (1-p)S = (1-p)^1 + (1-p)^2 + (1-p)^3 + \cdots = \frac{(1-p)}{p}$$

$$pS = \frac{(1-p)}{p} \implies S = \frac{(1-p)}{p^2}$$

$$E[X] = \sum_{j=0}^{\infty} j \cdot \Pr[X = j] = \sum_{j=1}^{\infty} j(1-p)^{j-1} p = \frac{p}{1-p} \sum_{j=1}^{\infty} j(1-p)^j$$

$$= \frac{p}{1-p} \cdot \frac{(1-p)}{p^2} = \frac{1}{p}$$

(13.8) Linearity of Expectation. Given two random variables $X$ and $Y$ defined over the same probability space, we can define $X+Y$ to be the random variable equal to $X(\omega) + Y(\omega)$ on a sample point $\omega$. For any $X$ and $Y$, we have

$$E[X+Y] = E[X] + E[Y].$$

(13.10) $H(n) = \sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ harmonic number $H(n)$.

$$\int_1^{n+1} \frac{1}{x} dx = \ln(n+1)$$

$\ln(n+1) < H(n) < 1 + \ln n$, and more loosely $H(n) = \Theta(\log n)$.

Conditional Expectation

Suppose we have a random variable $X$ and an event $\mathcal{E}$ of positive probability. Then we define the conditional expectation of $X$, given $\mathcal{E}$, to be the expected value of $X$ computed only over the part of sample space corresponding to $\mathcal{E}$.

$$E[X \mid \mathcal{E}] = \sum_{j=0}^{\infty} j \cdot Pr[X = j \mid \mathcal{E}].$$

## 13.4 A Randomized Approximation Algorithm for MAX 3-SAT

Algorithm: set each variable $x_1, \ldots, x_n$ independently to 0 or 1 with probability $\frac{1}{2}$ each.

let $Z_i = 1$ if clause $C_i$ is satisfied, and 0 otherwise.

Thus $Z = Z_1 + Z_2 + \cdots + Z_k$. $E[Z_i]$ is equal to the probability that $C_i$ is satisfied.

In order for $C_i$ not to be satisfied, each of its three variables must be assigned the value that fails to make it true; $(\frac{1}{2})^3 = \frac{1}{8}$ so $E[Z_i] = \frac{7}{8}$.

$$E[Z] = E[Z_1] + E[Z_2] + \cdots + E[Z_k] = \frac{7}{8}k.$$

(13.14) Consider a 3-SAT formula, where each clause has three different variables. The expected number of clauses satisfied by a random assignment is within an approximation factor $\frac{7}{8}$ of optimal.

(13.15) For every instance of 3-SAT, there is a truth assignment that satisfies at least a $\frac{7}{8}$ fraction of all clauses.

Cute application: Every instance of 3-SAT with at most seven clauses is satisfiable.

(13.16) There is a randomized algorithm with polynomial expected running time that is guaranteed to produce a truth assignment satisfying at least $\frac{7}{8}$ fraction of all clauses. $P \geq \frac{1}{8k}$ $n \leq 8k$

# Fibonacci Heaps [Introduction to Algorithms by CLRS]

dual purpose:

1) supports a set of operations that constitutes a "mergeable heap".
2) several operations run in constant amortized time.

mergeable heap:

- MAKE-HEAP()
- INSERT(H, x) inserts element x, whose key has already been filled in, into heap H.
- MINIMUM(H) returns a pointer to the element in heap H whose key is minimum.
- EXTRACT-MIN(H)
- UNION($H_1$, $H_2$)

Fibonacci heaps also support:

+ DECREASE-KEY(H, x, k) assigns to element x within H the new key value k, which we assume to be no greater than its current key value.
+ DELETE(H, x)

| Procedure | Binary heap (worst-case) | Fibonacci heap (amortized) |
|---|---|---|
| MAKE-HEAP | $\Theta(1)$ | $\Theta(1)$ |
| INSERT | $\Theta(\lg n)$ | $\Theta(1)$ |
| MINIMUM | $\Theta(1)$ | $\Theta(1)$ |
| EXTRACT-MIN | $\Theta(\lg n)$ | $O(\lg n)$ |
| UNION | $\Theta(n)$ | $\Theta(1)$ |
| DECREASE-KEY | $\Theta(\lg n)$ | $\Theta(1)$ |
| DELETE | $\Theta(\lg n)$ | $O(\lg n)$ |

desirable when number of these operations are small

application:

1) counting minimum spanning trees
2) single-source shortest paths.

drawbacks:

- large constant factors
- programming complexity

Fibonacci heap:

A Fibonacci heap is a collection of rooted trees that are min-heap ordered.

min-heap property:

the key of a node is greater than or equal to the key of its parent

## Potential function

$t(H)$ the number of trees in the root list
$m(H)$ the number of marked nodes in $H$.

$$\Phi(H) = t(H) + 2m(H)$$

- assume that a unit of potential can pay for a constant amount of work

## Maximum degree

assume that we know an upper bound $D(n)$ on the maximum degree of any node in an $n$-node Fibonacci heap.

$$D(n) \leq \lfloor \lg n \rfloor$$

when we support DECREASE-KEY and DELETE, $D(n) = O(\lg n)$.

## Inserting a node

- just add it to the root list
- the increase in potential $t(H') = t(H) + 1$, $m(H') = m(H)$ is 1, actual cost $O(1)$, amortized cost $O(1) + 1 = O(1)$

## Uniting two Fibonacci heaps

- change in potential

$$\Phi(H) - (\Phi(H_1) + \Phi(H_2))$$
$$= (t(H) + 2m(H)) - ((t(H_1) + 2m(H_1)) + (t(H_2) + 2m(H_2)))$$
$$= 0$$

The amortized cost of FIB-HEAP-UNION is equal to its $O(1)$ actual cost.

## Extracting the minimum node

```
FIB-HEAP-EXTRACT-MIN(H)
    z = H.min
    if z ≠ NIL
        for each child x of z
            add x to the root list of H
            x.p = NIL
        remove z from the root list of H
        if z == z.right
            H.min = NIL
        else H.min = z.right      ← not necessarily going to be the new minimum node.
            CONSOLIDATE(H)
        H.n = H.n - 1
    return z
```

```
CONSOLIDATE(H)
  let A[0 .. D(H.n)] be a new array  // keep track of roots according to their degrees
  for i = 0 to D(H.n)
      A[i] = NIL
  for each node w in the root list of H
      x = w
      d = x.degree
      while A[d] ≠ NIL
          y = A[d]          // another node with the same degree as x
          if x.key > y.key
              exchange x with y
          FIB-HEAP-LINK(H, y, x)
          A[d] = NIL
          d = d+1
      A[d] = x
  H.min = NIL
  for i = 0 to D(H.n)
      if A[i] ≠ NIL
          if H.min == NIL
              create a root list of H containing just A[i]
              H.min = A[i]
          else insert A[i] into H's root list
              if A[i].key < H.min.key
                  H.min = A[i]
```

need to know upper bound.

loop invariant: At the start of each iteration of the while loop $d = x.degree$

② the total amount of work in the for loop is at most proportional to $D(n) + t(H)$

```
FIB-HEAP-LINK(H, y, x)
1. remove y from the root list of H
2. make y a child of x, incrementing x.degree
3. y.mark = FALSE
```

amortized cost : try to show it is $O(D(n))$.

① $O(D(n))$ contribution comes from FIB-HEAP-EXTRACT-MIN processing at most $D(n)$ children of the minimum node.

Thus, the total actual work is ①② in $O(D(n) + t(H))$.

potential before extracting minimum $t(H) + 2m(H)$

potential afterwards is at most $(D(n)+1) + 2m(H)$

can scale up the units of potential to dominate the constant.

$$O(D(n) + t(H)) + ((D(n)+1) + 2m(H)) - (t(H) + 2m(H)) = O(D(n)) + O(t(H)) - t(H) = O(D(n))$$

Bounding the maximum degree

to show the upper bound of $D(n)$ is $O(\lg n)$.

In particular, $D(n) \le \lfloor \log_\phi n \rfloor$

$\phi = \frac{1+\sqrt{5}}{2}$

Lemma 19.1

Let $x$ be any node in a Fibonacci heap, and suppose that $x.degree = k$. Let $y_1, y_2, \dots, y_k$ denote the children of $x$ in the order in which they were linked to $x$ from the earliest to the latest. Then $y_1.degree \ge 0$ and $y_i.degree \ge i-2$ for $i = 2, 3, \dots k$.

Lemma 19.2

For all integers $k \ge 0$,

$$F_{k+2} = 1 + \sum_{i=0}^{k} F_i \qquad F_k = \begin{cases} 0 & , k=0 \\ 1 & , k=1 \\ F_{k-1} + F_{k-2} & , k \ge 2 \end{cases}$$

Lemma 19.3

For all integers $k \ge 0$, the $(k+2)$nd Fibonacci number satisfies $F_{k+2} \ge \phi^k$

inductive step:
$$\begin{aligned} F_{k+2} &= F_{k+1} + F_k \\ &\ge \phi^{k-1} + \phi^{k-2} \quad \text{(by the inductive hypothesis)} \\ &= \phi^{k-2}(\phi + 1) \\ &= \phi^{k-2} \cdot \phi^2 \quad (\phi \text{ is the positive root of equation } x^2 = x+1) \\ &= \phi^k \end{aligned}$$

Lemma 19.4

Let $x$ be any node in a Fibonacci heap, and let $k = x.degree$. Then $size(x) \ge F_{k+2} \ge \phi^k$, where $\phi = \frac{1+\sqrt{5}}{2}$

Proof. let $s_k$ denote minimum possible size of any node of degree $k$ in any Fibonacci heap.

$s_k \le size(x)$

$$\begin{aligned} size(x) &\ge s_k \\ &\ge 2 + \sum_{i=2}^{k} s_{y_i.degree} \\ &\ge 2 + \sum_{i=2}^{k} s_{i-2} \\ &\ge 2 + \sum_{i=2}^{k} F_i \\ &= 1 + \sum_{i=0}^{k} F_i \\ &= F_{k+2} \quad \text{(by Lemma 19.2)} \\ &\ge \phi^k \quad \text{(by Lemma 19.3)} \end{aligned}$$

$\therefore size(x) \ge s_k \ge F_{k+2} \ge \phi^k$.

Corollary 19.5

The maximum degree $D(n)$ of any node in an $n$-node Fibonacci heap is $O(\lg n)$.

$n \ge size(x) \ge \phi^k$ where $k = x.degree$

$k \le \lfloor \log_\phi n \rfloor$

# Maximum Flows and Parametric Shortest Paths in Planar Graphs

by Jeff Erickson jeffe@cs.uiuc.edu

let $G=(V,E)$ be a directed plane graph. s.t be vertices of $G$

let $c: E \longrightarrow \mathbb{R}$ be a nonnegative capacity function

Goal: compute a maximum $(s,t)$-flow in $G$.

Assume WLOG the reversal of any directed edge in $G$ is also an edge in $G$

$\longrightarrow$ implies that both $G$ and its dual $G^*$ are strongly connected.

## Venkatesan's Reduction

Idea: compute a feasible $(s,t)$-flow with fixed value $\lambda$, or correctly report that no such flow exists, by reduction to a single source shortest path problem in an appropriately weighted dual graph $G^*$.

$\pi(e)$: Fixed an arbitrary directed path $P$ from $s$ to $t$, and let $\pi: E \to \mathbb{R}$ denote the unit flow through $P$:

$$\pi(e) := \begin{cases} 1 & \text{if } e \in P \\ -1 & \text{if } \operatorname{rev}(e) \in P \\ 0 & \text{otherwise} \end{cases}$$

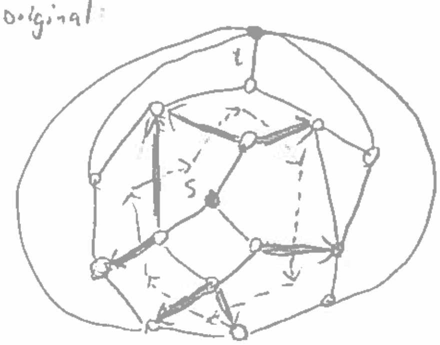
For any subset $E' \subseteq E$, let $\pi(E') = \sum_{e \in E'} \pi(e)$.

cocycle: A subgraph $C$ of $G$ is called a cocycle if the corresponding dual subgraph $C^*$ is a simple directed cycle in $G^*$.

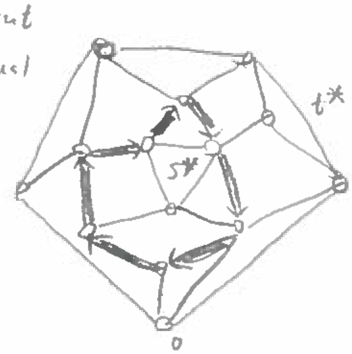
crossing number: For any cycle $C^*$ in $G^*$, we call $\pi(C)$ the crossing number of $C^*$.

Lemma 2.1 $\pi(C) \in \{-1, 0, 1\}$ for any cocycle $C$. Moreover, $\pi(C) = 1$ if and only if $C$ is an $(s,t)$-cut

original:




dual



consider the flow $\lambda \cdot \pi$, which assigns value $\begin{cases} \lambda & \text{to every directed edge in path } P \\ -\lambda & \text{" " " " " } rev(P) \\ 0 & \text{to every other edge.} \end{cases}$

Let $G_\lambda := G_{\lambda \cdot \pi}$ denote the residual network of this flow

capacity function: $c(\lambda, e) := c(e) - \lambda \cdot \pi(e)$.

$\therefore$ $\lambda \cdot \pi$ is feasible if and only if $c(\lambda, e) \geq 0$ for every edge $e$ in $G$.

Let $G_\lambda^*$ denote the dual residual network, which is the directed dual graph $G^*$ where every edge $e^*$ has a cost $c(\lambda, e^*) = \underline{c(\lambda, e)}$.

**Lemma 2.2** There is a feasible $(s,t)$-flow in $G$ with value $\lambda$ if and only if the residual network $G_\lambda^*$ does not contain a negative cycle.

let $dist(\lambda, p)$ denote the shortest path distance in $G_\lambda^*$ from $o$ to $p$. (arbitrary dual vertex called origin)

define $\phi(\lambda, e) := dist(\lambda, head(e^*)) - dist(\lambda, tail(e^*)) + \lambda \cdot \pi(e)$.

because the duals of the edges leaving $v$ define a directed cycle in $G^*$, all the $dist(\lambda, \cdot)$ terms in the sum cancel out.

$\therefore$ $\phi(\lambda, \cdot)$ is a valid $(s,t)$-flow with value $\lambda$.

define the <u>slack</u> of each dual edge $e^*$

$slack(\lambda, e^*) := dist(\lambda, tail(e^*)) + c(\lambda, e) - dist(\lambda, head(e^*))$

so $slack(\lambda, e^*) = c(e) - \phi(\lambda, e)$

## Parametric Shortest Paths

Let $\lambda_{max}$ denote the largest value of $\lambda$ for which shortest paths in $G_\lambda^*$ are well-defined. Lemma 2.2 implies that $\lambda_{max}$ is also the value of the maximum flow.

For any particular value of $\lambda$, let $\underline{T_\lambda}$ denote the single-source shortest path tree in $G_\lambda^*$ rooted at $o$.

### High level algorithm

> PLANARMAXFLOW $(G, c, s, t)$:
> Compute $T_0$
> Maintain $T_\lambda$ as $\lambda$ increases continuously from $0$ to $\lambda_{max}$
> Compute $\phi(\lambda_{max}, \cdot)$ from $T_{\lambda_{max}}$

## Genericity assumption

we assume that the capacity function is generic.

1) Our genericity assumption implies that $T_\lambda$ is uniquely defined for all $\lambda$ between $0$ and $\lambda_{max}$, except for a finite set of critical values

2) Our genericity assumption implies that exactly one non-tree edge becomes tense at each critical value of $\lambda$.

**Lemma 2.3** $\lambda_{max}$ is the first critical value of $\lambda$ whose pivot introduces a directed cycle into $T_\lambda$.

**pivot** At each critical value of $\lambda$, some non-tree dual edge $p \to q$ becomes tense and enters $T_\lambda$, replacing the previous edge $pred(\lambda, q) \to q$.

**tense** call a dual edge $e^*$ tense if $slack(\lambda, e^*) = 0$.

**Lemma 2.4** $\lambda_{max}$ is the smallest critical value of $\lambda$ whose pivot disconnects $L_\lambda$

**loose** call a primal edge $e$ loose at $\lambda$ if neither its dual $e^*$ nor its reversed dual $rev(e^*)$ is tense at $\lambda$, and let $L_\lambda$ be the graph of all loose edges.

**active** A dual edge is active at $\lambda$ if its slack at $\lambda$ is decreasing

**$LP_\lambda$** The primal spanning tree $L_\lambda$ contains a unique directed path from $s$ to $t$; call this loose path $LP_\lambda$.

**Lemma 2.5** A dual edge $e^*$ is active at $\lambda$ if and only if $e$ is an edge of $LP_\lambda$.

☆ **PLANARMAXFLOW$(G, c, s, t)$:**

```
Initialize the spanning tree L, predecessors and slacks
while s and t are in the same component of L
        LP ← the path from s to t
        p→q ← the edge in LP* with minimum slack
          Δ ← slack(p→q)
        for every edge e in LP
              slack(e*) ← slack(e*) − Δ
              slack(rev(e*)) ← slack(rev(e*)) + Δ
        delete (p→q)* from L
        if q≠o  << that is, if pred(q) ≠ ∅ >>
              insert (pred(q)→q)* into L
        pred(q) ← p
for each edge e
        φ(e) ← c(e) − slack(e*)
```

} maintain $T_\lambda$ as $\lambda$ increases by pivoting until $\lambda_{max}$

} Compute the flow