Strong Bridges and Strong Articulation Points of Directed Graphs

Giuseppe F. Italiano

Univ. of Rome "Tor Vergata"

Based on joint work with Donatella Firmani, Luigi Laura, Alessio Orlandi and Federico Santaroni

Outline of the Talk

1. Preliminary Definitions
2. Algorithms for strong articulation points and strong bridges
3. Preliminary Experiments on Large Scale Graphs
4. Conclusions

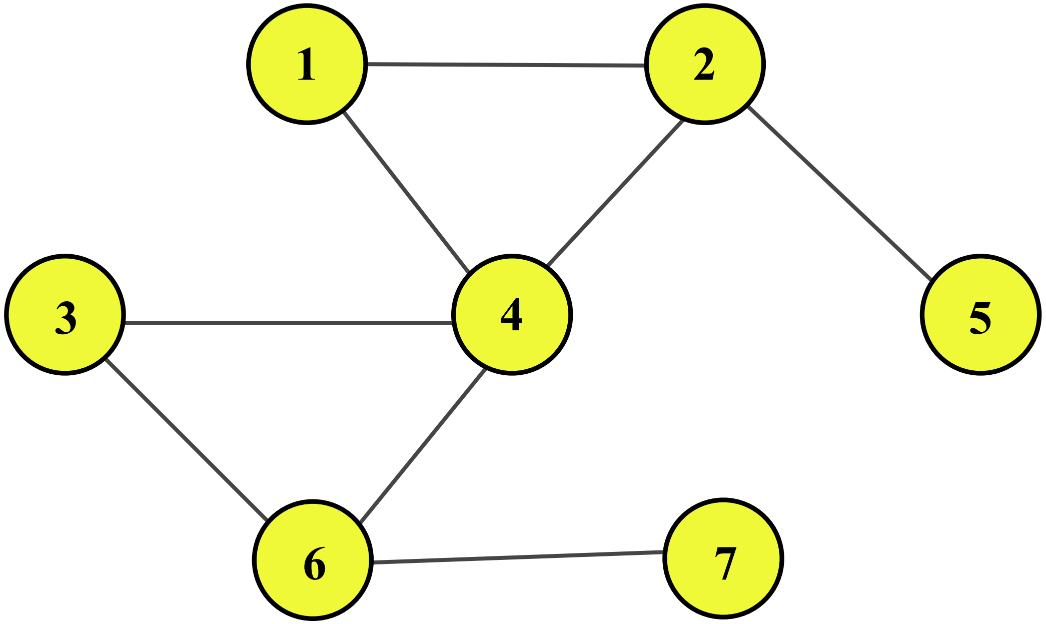
2-Edge Connectivity

Let $G = (V,E)$ be an **undirected** connected graph, with m edges and n vertices.

An edge $e \in E$ is a **bridge** if its removal disconnects G (i.e., increases the number of connected components of G)

Graph G is **2-edge-connected** if it has no bridges.

The **2-edge-connected components** of G are its maximal 2-edge-connected subgraphs



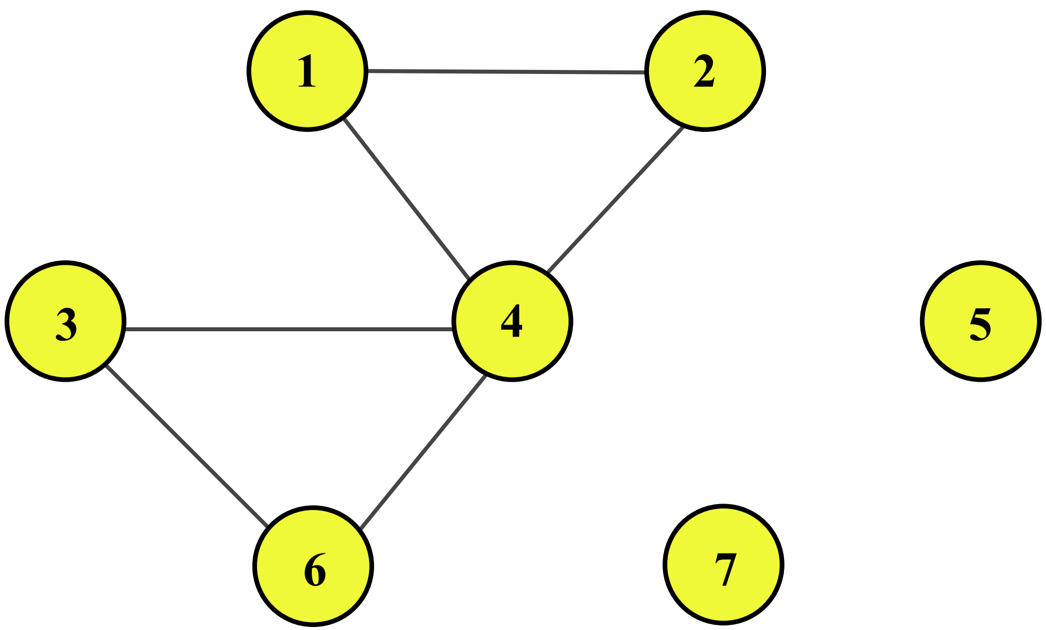
2-Edge Connectivity

Let *G* = *(V,E)* be an **undirected** connected graph, with *m* edges and *n* vertices.

An edge $e \in E$ is a **bridge** if its removal disconnects *G* (i.e., increases the number of connected components of *G*)

Graph *G* is **2-edge-connected** if it has no bridges.

The **2-edge-connected components** of *G* are its maximal 2-edge-connected subgraphs



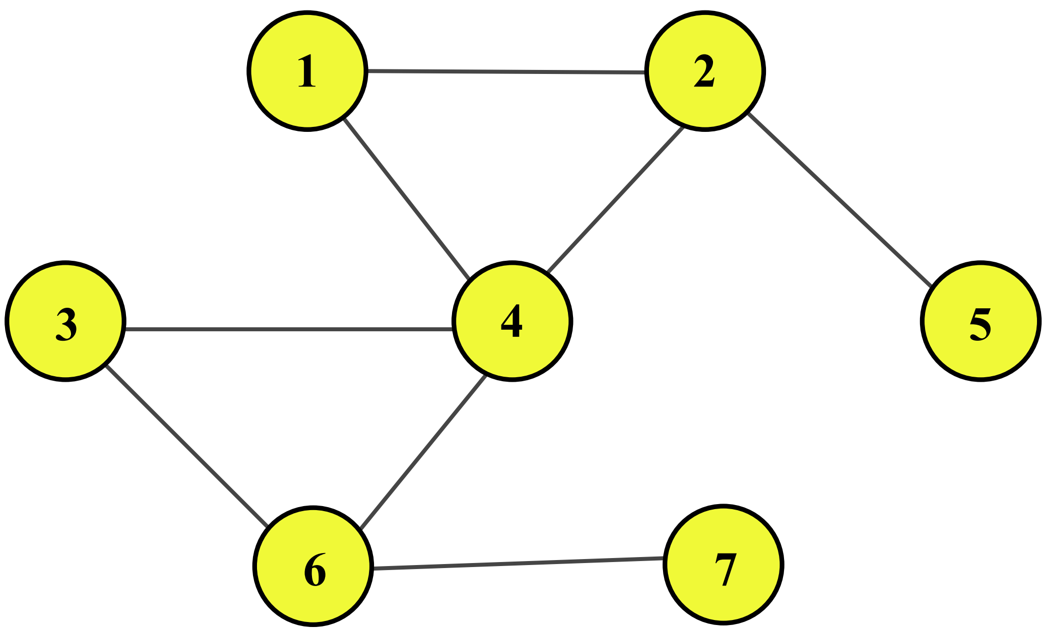
2-Vertex Connectivity

Let $G = (V,E)$ be an **undirected** connected graph, with m edges and n vertices.

A vertex $v \in V$ is an **articulation point** if its removal disconnects G (i.e., increases the number of connected components of G)

Graph G is **2-vertex-connected** if it has no articulation points.

The **2-vertex-connected components** of G are its maximal 2-vertex-connected subgraphs



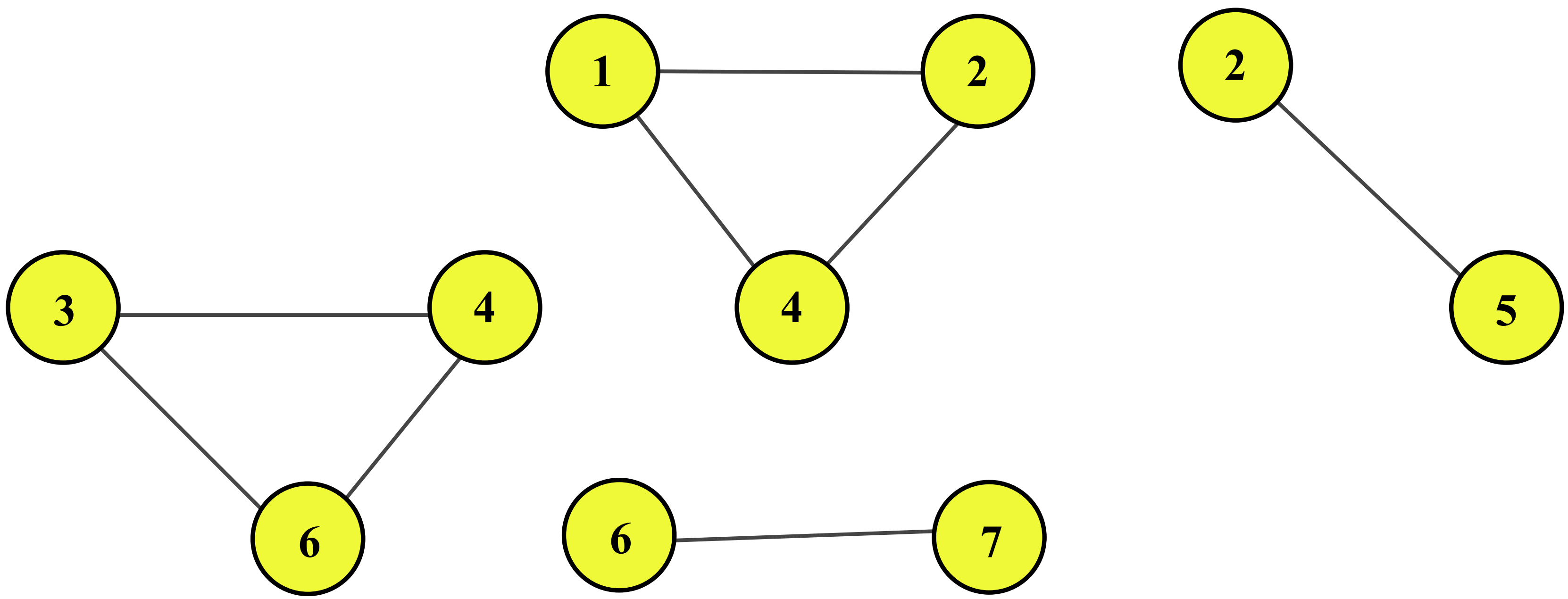
2-Vertex Connectivity

Let $G = (V,E)$ be an **undirected** connected graph, with m edges and n vertices.

A vertex $v \in V$ is an **articulation point** if its removal disconnects G (i.e., increases the number of connected components of G)

Graph G is **2-vertex-connected** if it has no articulation points.

The **2-vertex-connected components** of G are its maximal 2-vertex-connected subgraphs



Bounds for Undirected *G*

Q1: Find whether *G* is 2-vertex-connected (2-edge-connected).
I.e., find **one** connectivity cut (if any) ***O(m+n)***

Q2: Find **all** connectivity cuts (articulation points, bridges) in *G* ***O(m+n)***

Q3: Find the **connectivity** (2-vertex-, 2-edge-connected) **components** of *G* ***O(m+n)***

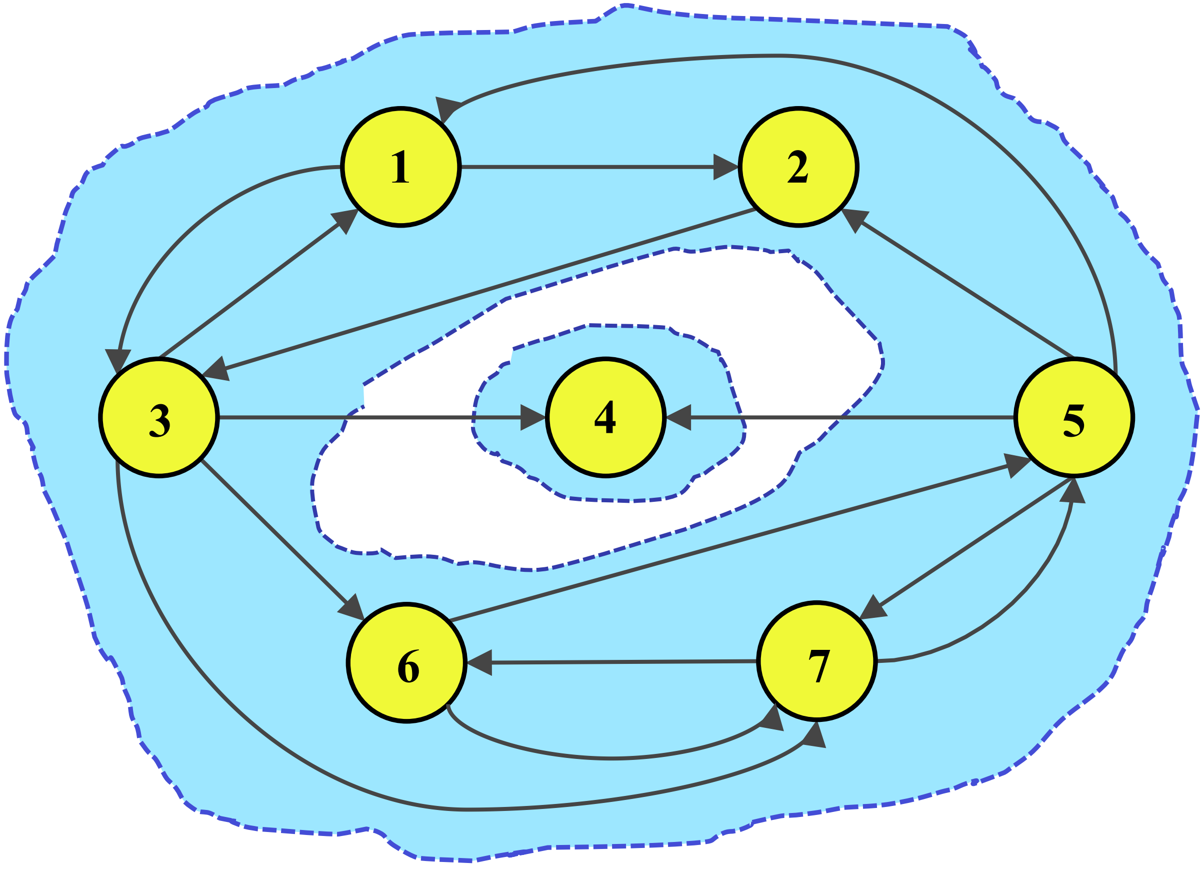
[Hopcroft & Tarjan 1973], [Tarjan 1974]

Directed Graphs

Let $G = (V,E)$ be a **directed** graph, with m edges and n vertices.

G is **strongly connected** if there is a directed path from each vertex to every other vertex in G.

The **strongly connected components** (SCCs) of G are its maximal strongly connected subgraphs.



Directed: 2-Vertex Connectivity

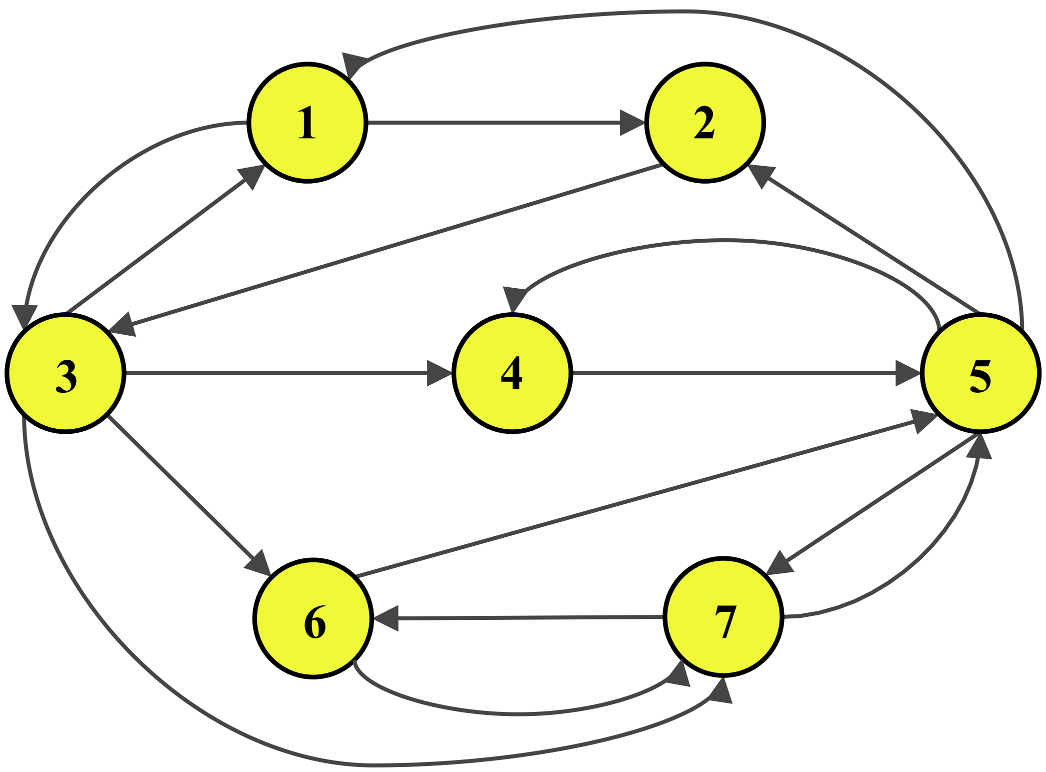
Let $G = (V,E)$ be a ***directed strongly*** connected graph, with m edges and n vertices.

A vertex $v \in V$ is a ***strong*** **articulation point** if its removal increases the number of ***strongly*** connected components of G

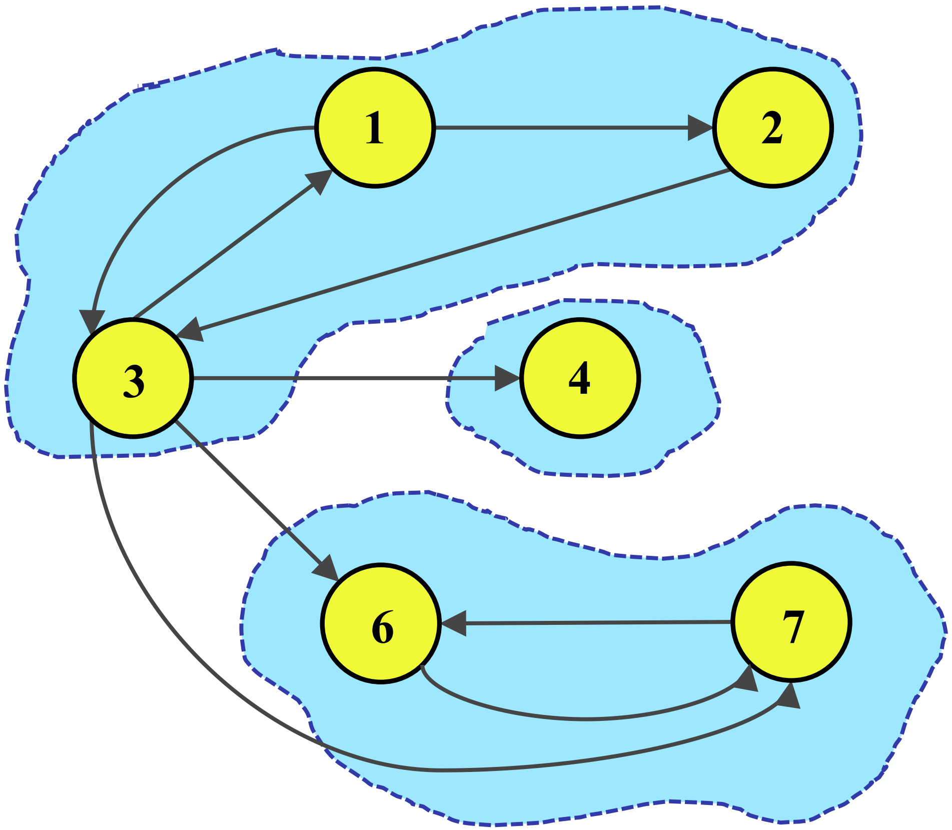
Graph G is **2-vertex-connected** if it has no ***strong*** articulation points.

The **2-vertex-connected components** of G are its maximal 2-vertex-connected subgraphs

Strong Articulation Points

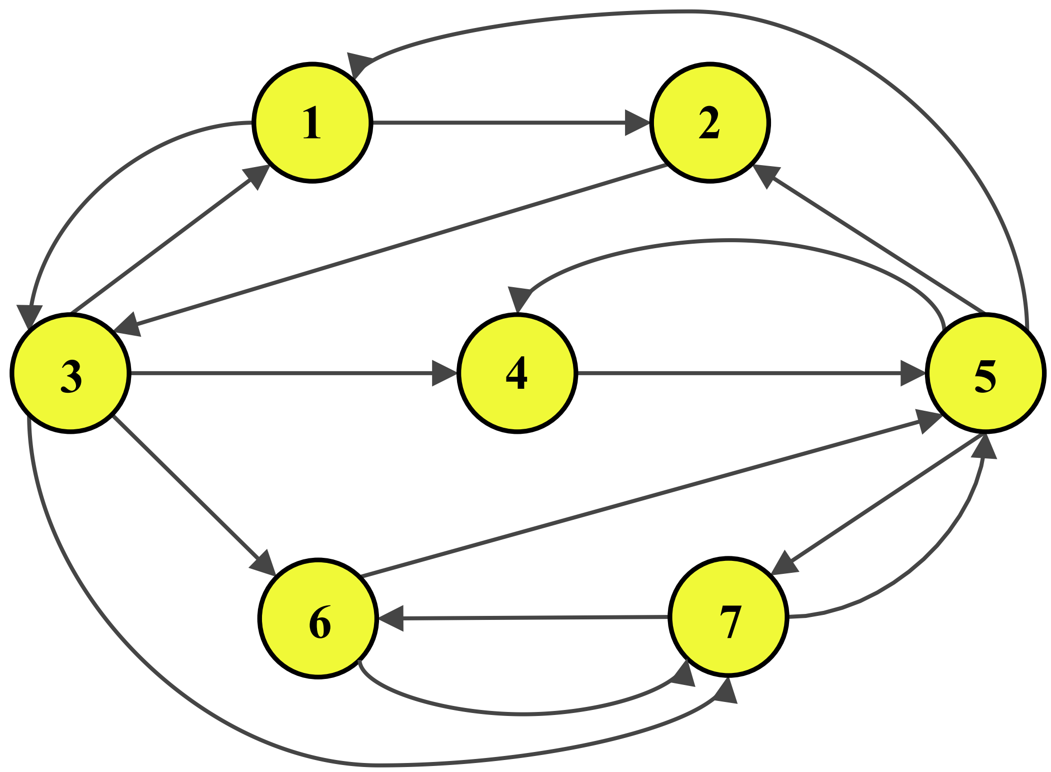


Strong Articulation Points

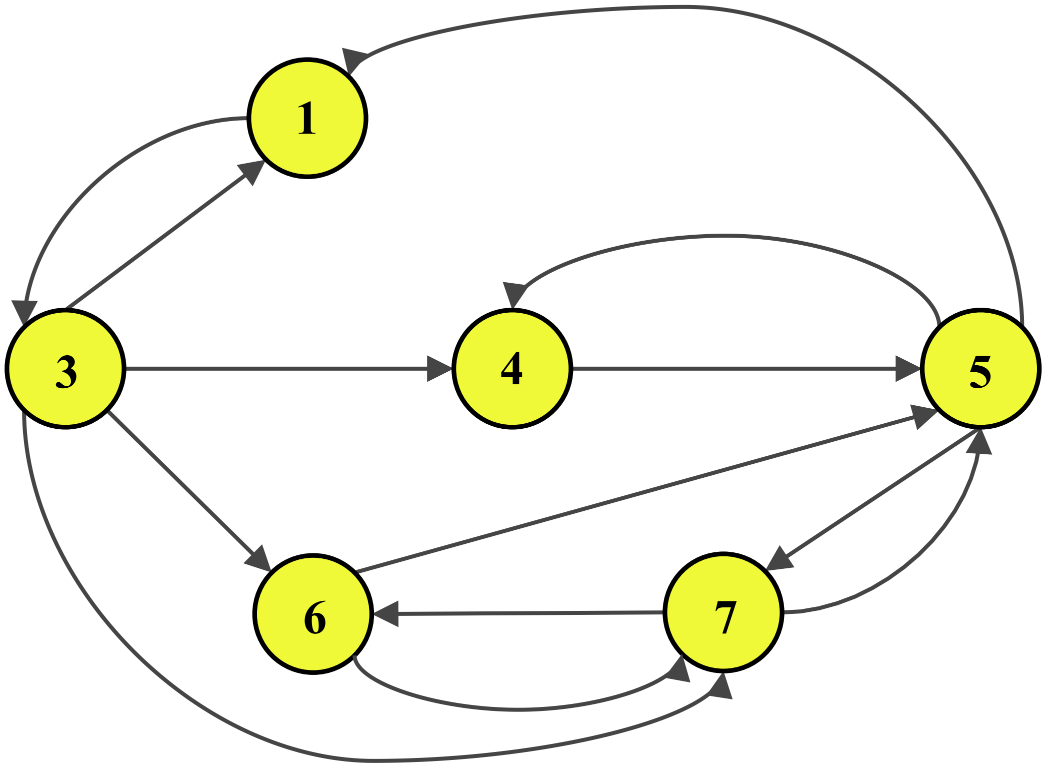


Vertex **5** is a strong articulation point

Strong Articulation Points

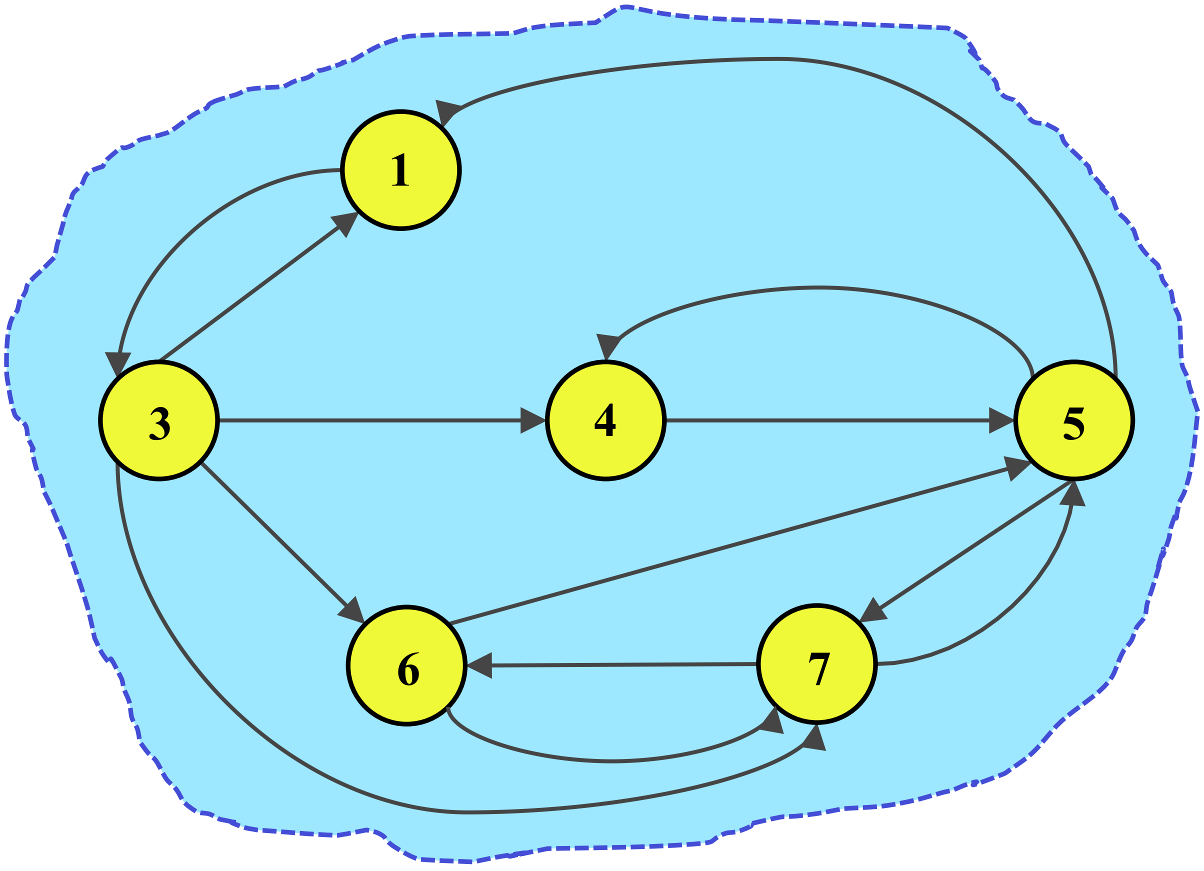


Strong Articulation Points



Strong Articulation Points

Vertex **2** is NOT a strong articulation point



Directed: 2-Edge Connectivity

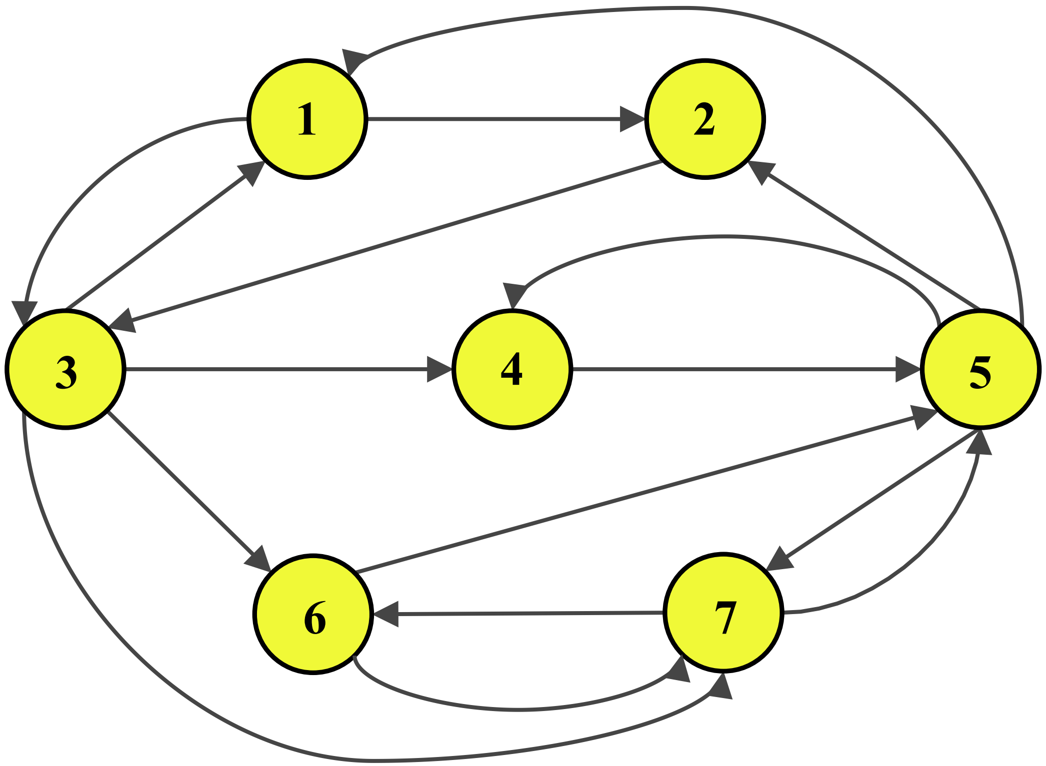
Let $G = (V,E)$ be a ***directed strongly*** connected graph, with m edges and n vertices.

An edge $(u,v) \in E$ is a ***strong*** **bridge** if its removal increases the number of ***strongly*** connected components of G

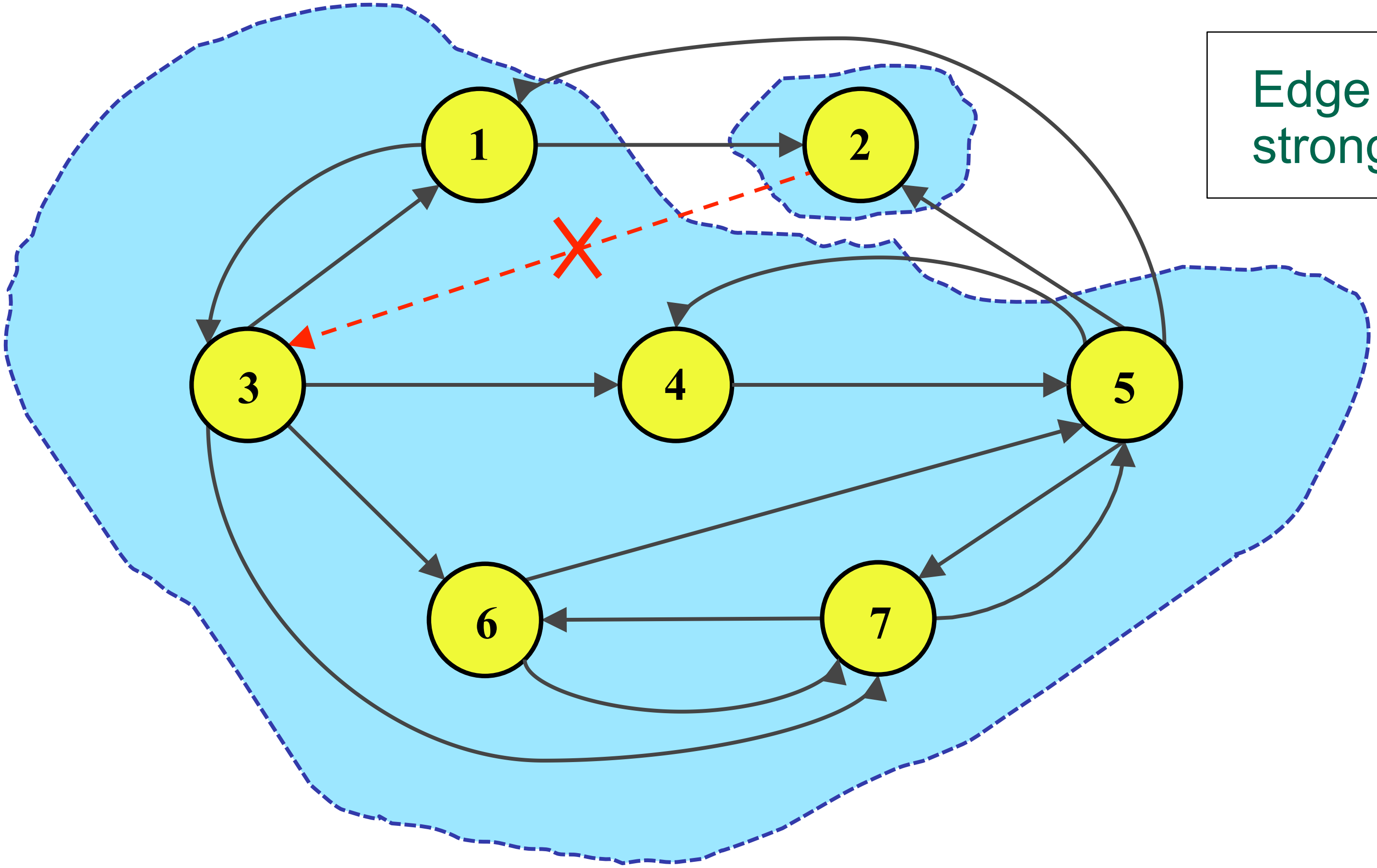
Graph G is **2-edge-connected** if it has no ***strong*** bridges.

The **2-edge-connected components** of G are its maximal 2-edge-connected subgraphs

Strong Bridges



Strong Bridges



Edge **(2,3)** is a strong bridge

Bounds for Directed *G*

Q1: Find whether directed *G* is 2-vertex-connected (2-edge-connected). I.e., find **one** connectivity cut (if any)

O(m+n)

[Tarjan 76] +
[Gabow & Tarjan 83]

[Georgiadis 10]

Q2: Find **all** connectivity cuts (articulation points, bridges) in *G*

O(m+n)

[Italiano et al 10]

Q3: Find the **connectivity** (2-vertex-, 2-edge-connected) **components** of *G*

?????

Further Motivation

Constraint programming - filtering for tree constraint: computing *all* strong articulation points open problem posed by Beldiceanu et al [2005]

Reliability in directed networks

Connectivity and flow of information in social networks [Mislove et al. 2007]

Speed up computation of matrix determinants [Bini & Pan 1994] [Maybee et al. 1989]

…

Outline of the Talk

1. Preliminary Definitions
2. **Algorithms for strong articulation points and strong bridges**
3. Preliminary Experiments on Large Scale Graphs
4. Conclusions

Naive Algorithms

Check whether vertex v is strong articulation point in G :

Compute strongly connected components of $G/\{v\}$

$O(n(m+n))$ for computing all strong articulation points

Check whether edge e is strong bridge in G :

Compute strongly connected components of $G/\{e\}$

$O(m(m+n))$ for computing all strong bridges

Not difficult to get $O(n(m+n))$ algorithm

Redundant Edges

Given a directed graph $G = (V,E)$, we say that an edge (u,v) is *redundant* if there is an alternative path from vertex u to vertex v avoiding edge (u,v). Otherwise, we say that (u,v) is *non-redundant*.

Observation. Let $G = (V , E)$ be a strongly connected graph. Then the edge $(u,v) \in E$ is a strong bridge if and only if (u,v) is non-redundant in G.

Computing strong bridges equivalent to computing redundant edges.

Boolean Matrix Multiplication?

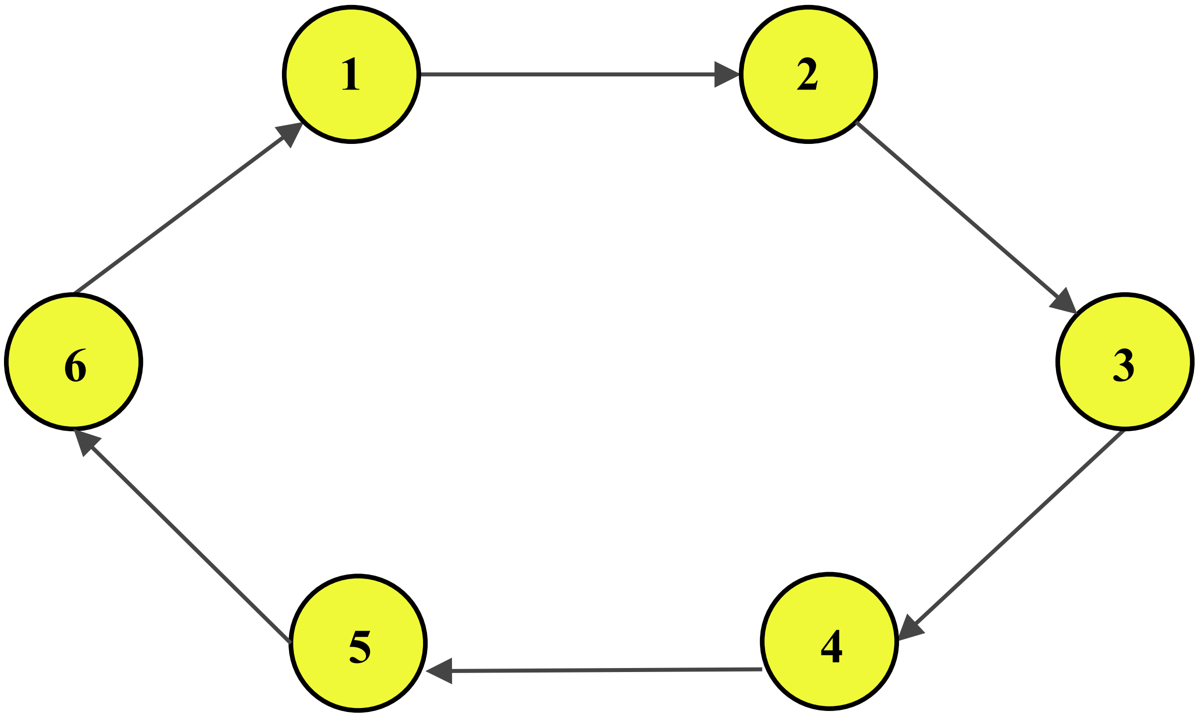
In a directed acyclic graph, finding all redundant edges is the *transitive reduction* problem.

Transitive reduction equivalent to *transitive closure* [Aho, Garey & Ullman 72]

Transitive closure equivalent to *Boolean matrix multiplication* [Furman 70], [Fischer & Meyer 71]

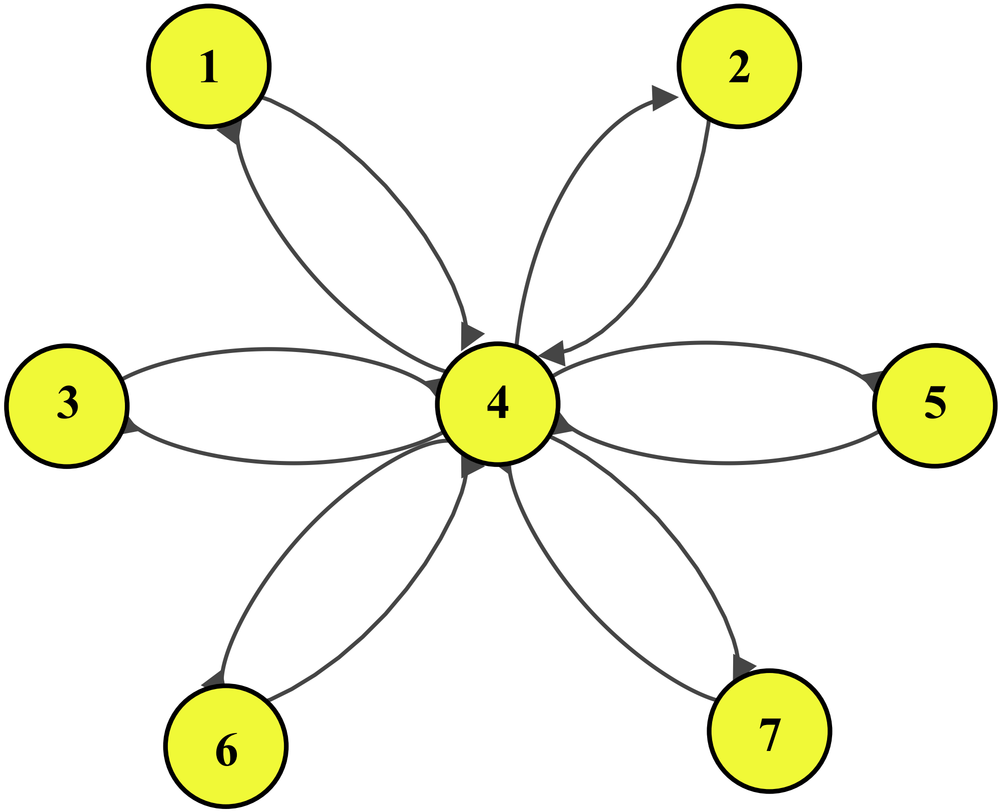
Thus, for DAGs the best known bound for computing redundant edges is $O(n^\omega)$.

Warm Up: How many SAP?



At most *n*

How Many Strong Bridges?



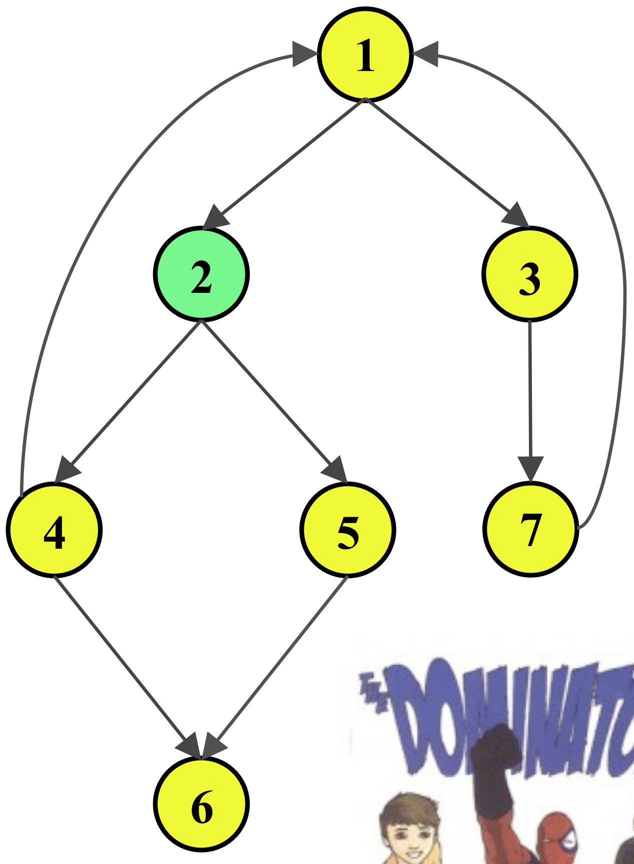
At most $2n-2$
(will prove it later)

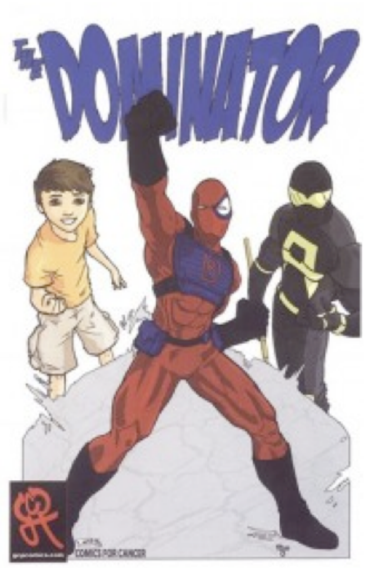
Vertex Dominators

A *flowgraph G(s) = (V,E,s)* is a directed graph with a start vertex s in V such that every vertex in V reachable from s

Given a flowgraph *G(s)=(V,E,s)*, can define a *dominance relation*: vertex *u* is *dominator* of vertex *v* if every path from *s* to *v* includes *u*

Let *dom(v)* be set of dominators of *v*. For any $v \neq s$ we have that $\{s,v\} \subseteq dom(v)$: *s* and *v* are the *trivial dominators* of *v*





Dominator Trees

Dominance relation is transitive and its transitive reduction is referred to as the dominator tree *DT(s)*.

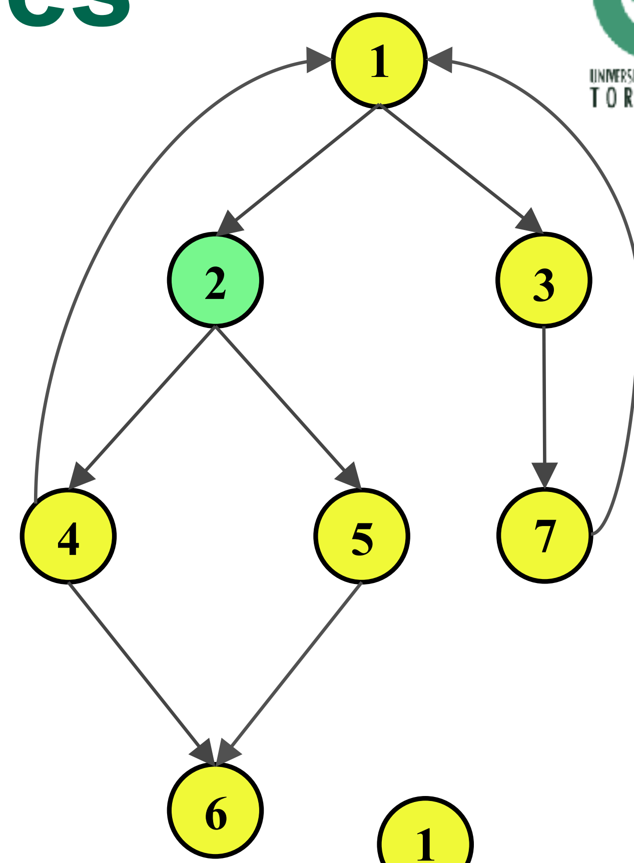
DT(s) rooted at *s*.

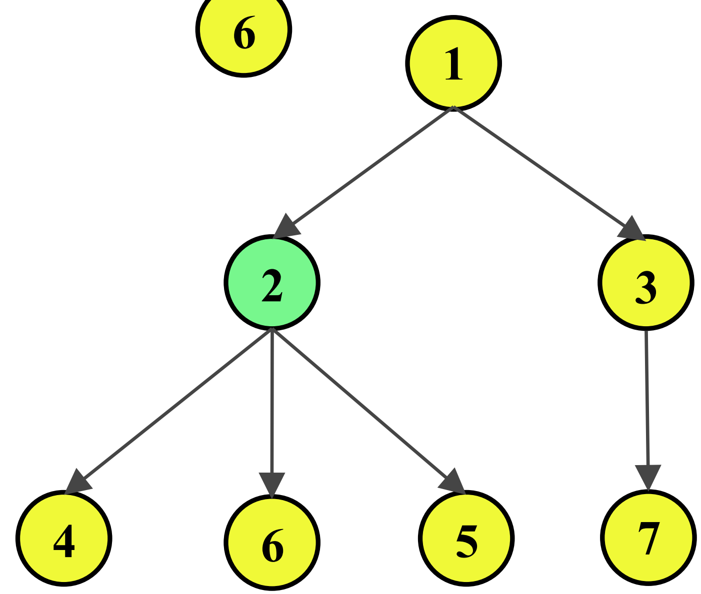
u dominates *v* if and only if *u* is ancestor of *v* in *DT(s)*.

If *u* is dominator of *v*, and every other non-trivial dominator of *u* also dominates *v*, *u* is an immediate dominator of *v*.

If *v* has any non-trivial dominators, then *v* has a unique immediate dominator: the immediate dominator of *v* is the parent of *v* in the dominator tree *DT(s)*.

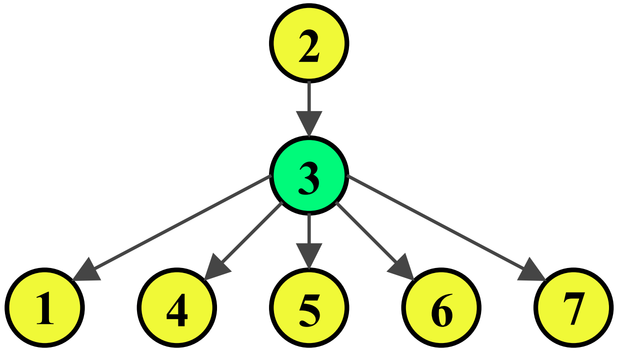
Dominators (and dominator tree) can be computed in *O(m+n)* time [Buchsbaum et al 1998]

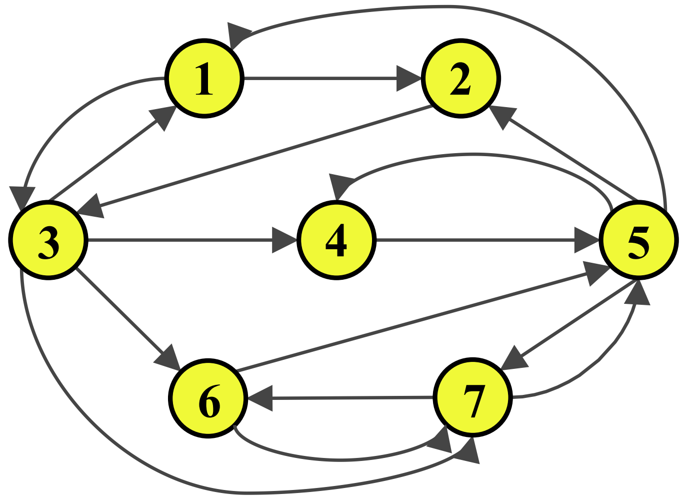




Vertex Dominators and SAP

Lemma 1 *Let G = (V,E) be a strongly connected graph, and let s be any vertex in G. Let G(s) = (V,E,s) be the flowgraph with start vertex s. If u is a non-trivial dominator of a vertex v in G(s), then u is a strong articulation point in G.*



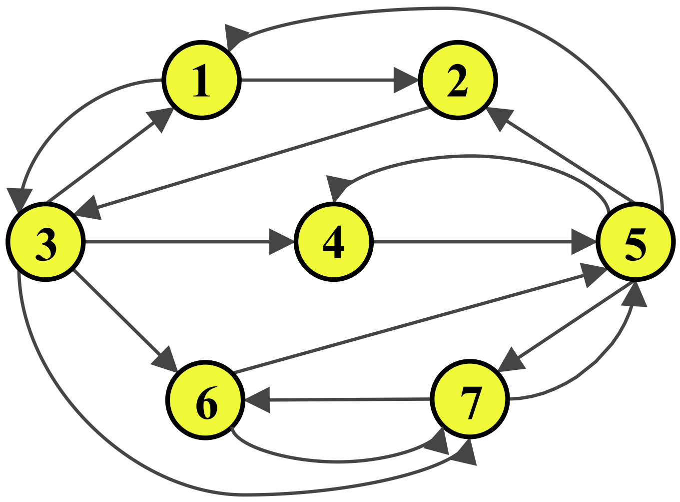


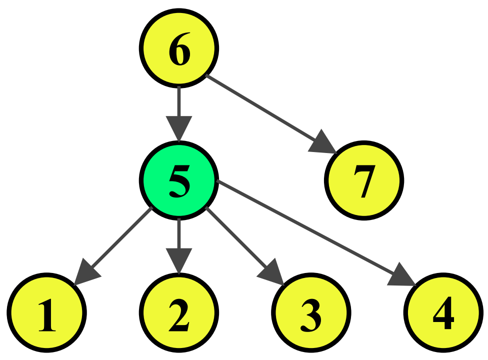
Vertex 3 is a non-trivial dominator in G(2)

Vertex 3 is strong articulation point in G

Vertex Dominators and SAP

Lemma 2 *Let G = (V,E) be a strongly connected graph. If u is a strong articulation point in G, then there must be a vertex s ∈ V such that u is a non-trivial dominator of a vertex v in the flowgraph G (s) = (V,E,s).*





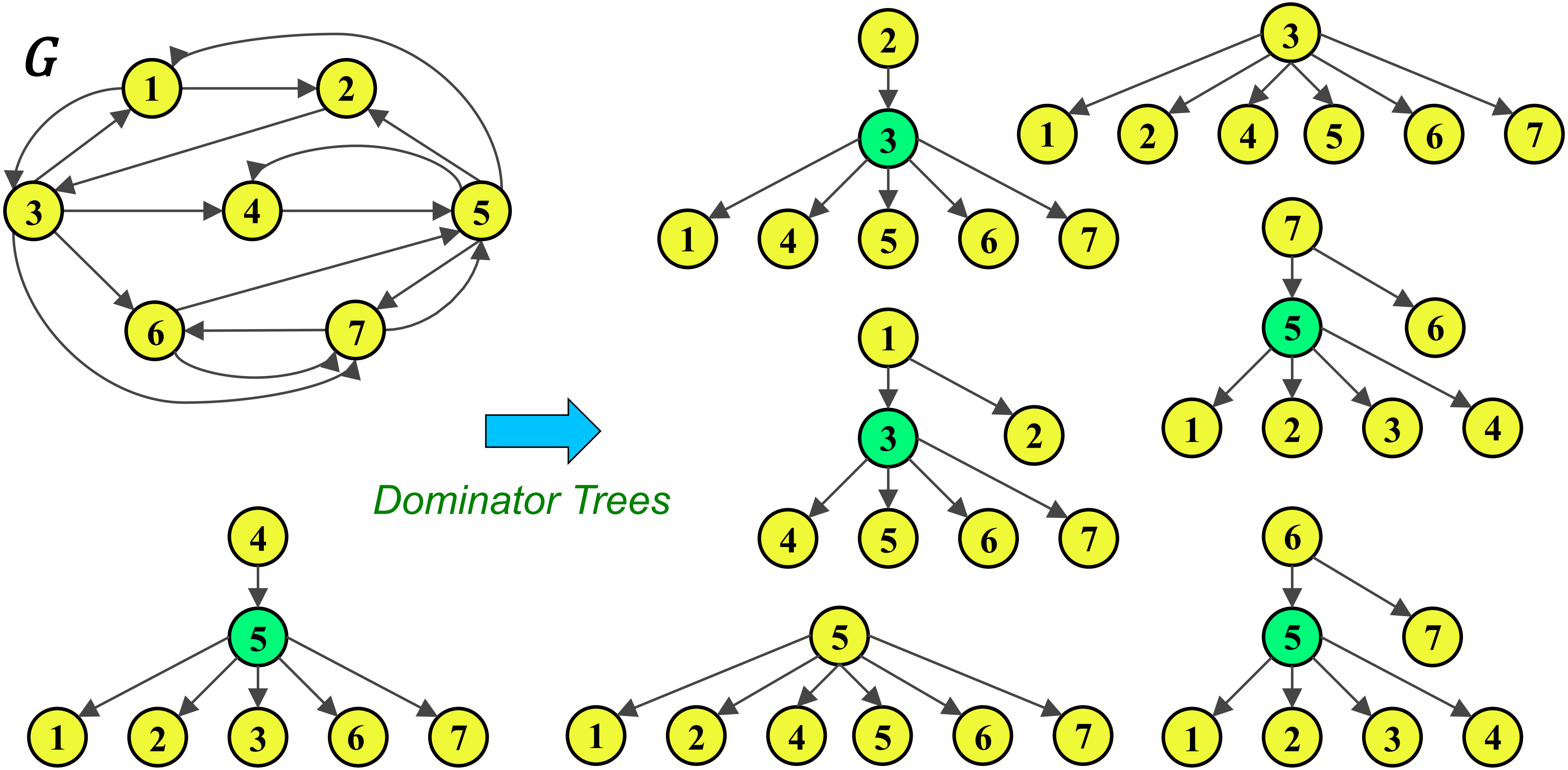
Vertex 5 is strong articulation point in G → ***Vertex 5 must be non-trivial dominator in some G(s). Here s=6.***

Still Not Efficient

Corollary *Let G = (V,E) be a strongly connected graph. Vertex u is a strong articulation point in G if and only there is a vertex s*$\in$*V such that u is a non-trivial dominator of a vertex v in the flowgraph G(s) = (V,E,s).*

Must compute dominator trees for all flowgraphs G(v), for each vertex v in V, and output all non-trivial dominators found.

Dominator Trees



Still Not Efficient

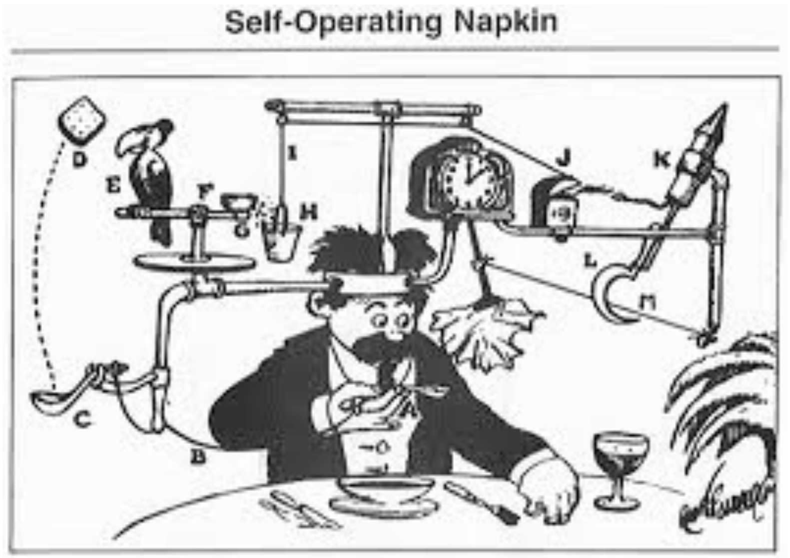
Corollary *Let G = (V,E) be a strongly connected graph. Vertex u is a strong articulation point in G if and only then there is a vertex s∈V such that u is a non-trivial dominator of a vertex v in the flowgraph G(s) = (V,E,s).*

Must compute dominator trees for all flowgraphs G(v), for each vertex v in V, and output all non-trivial dominators found.

Takes *O(n(m+n))* time

Like trivial algorithm

Only more complicated...



Reversal Graph

Reversal Graph $G^R = (V,E^R)$: reverse all edges in G.
If (u,v) in G then (v,u) in $G^{R.}$

Observation. *Let $G = (V,E)$ be a strongly connected graph and $G^R = (V,E^R)$ be its reversal graph. Then G^R is strongly connected. Furthermore, vertex v is a strong articulation point in G if and only if v is a strong articulation point in G^R.*

Exploit Dominators

Given a strongly connected graph $G=(V,E)$, let

- $G(s) = (V,E,s)$ be the flowgraph with start vertex s
- $D(s)$ the set of non-trivial dominators in $G(s)$
- $G^R(s) = (V,E^R,s)$ be the flowgraph with start vertex s
- $D^R(s)$ the set of non-trivial dominators in $G^R(s)$

Theorem. *Let $G = (V,E)$ be a strongly connected graph, and let $s \in V$ be any vertex in G. Then vertex $v \neq s$ is a strong articulation point in G if and only if $v \in D(s) \cup D^R(s)$.*

Strong Articulation Points

Theorem. *Let $G = (V,E)$ be a strongly connected graph, and let $s \in V$ be any vertex in G. Then vertex $v \neq s$ is a strong articulation point in G if and only if $v \in D(s) \cup D^R(s)$.*

Proof:

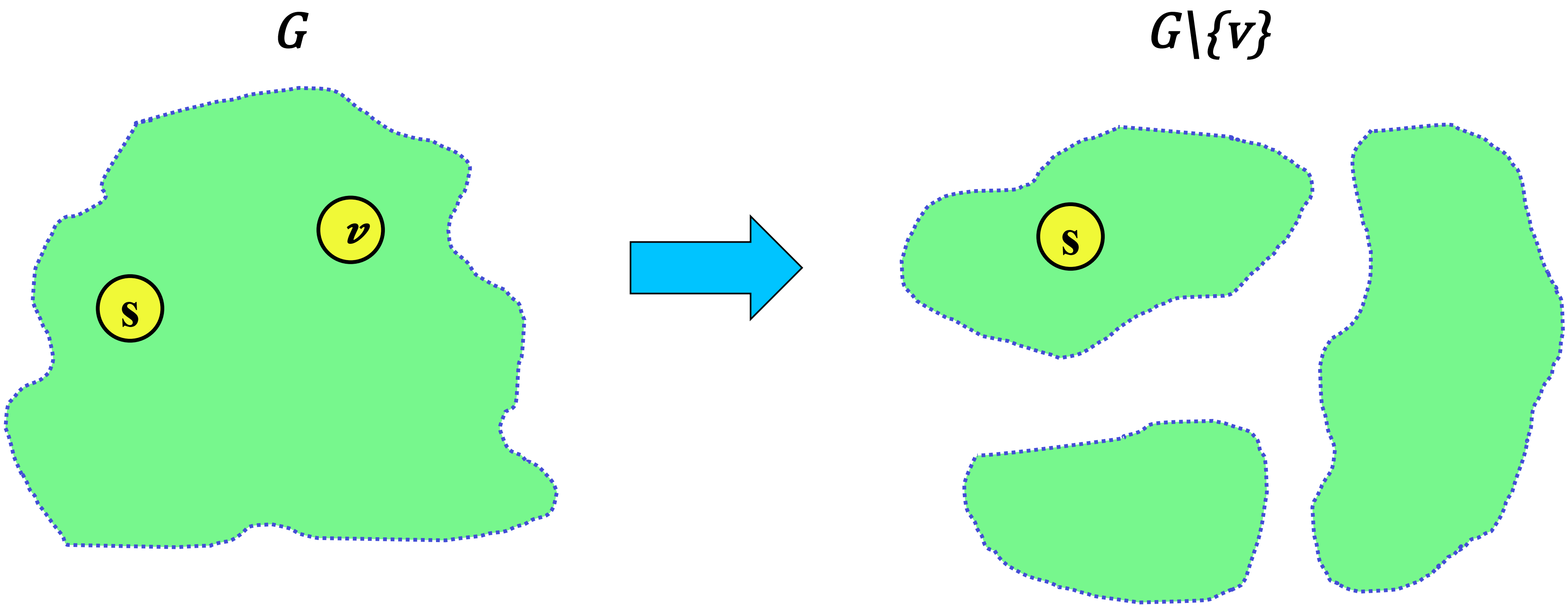
If $v \in D(s) \cup D^R(s)$ we know from previous lemmas that v must be an articulation point.

So, we need to prove only one direction.

Strong Articulation Points

Theorem. *Let $G = (V,E)$ be a strongly connected graph, and let $s \in V$ be any vertex in G. Then vertex $v \neq s$ is a strong articulation point in G if and only if $v \in D(s) \cup D^R(s)$.*

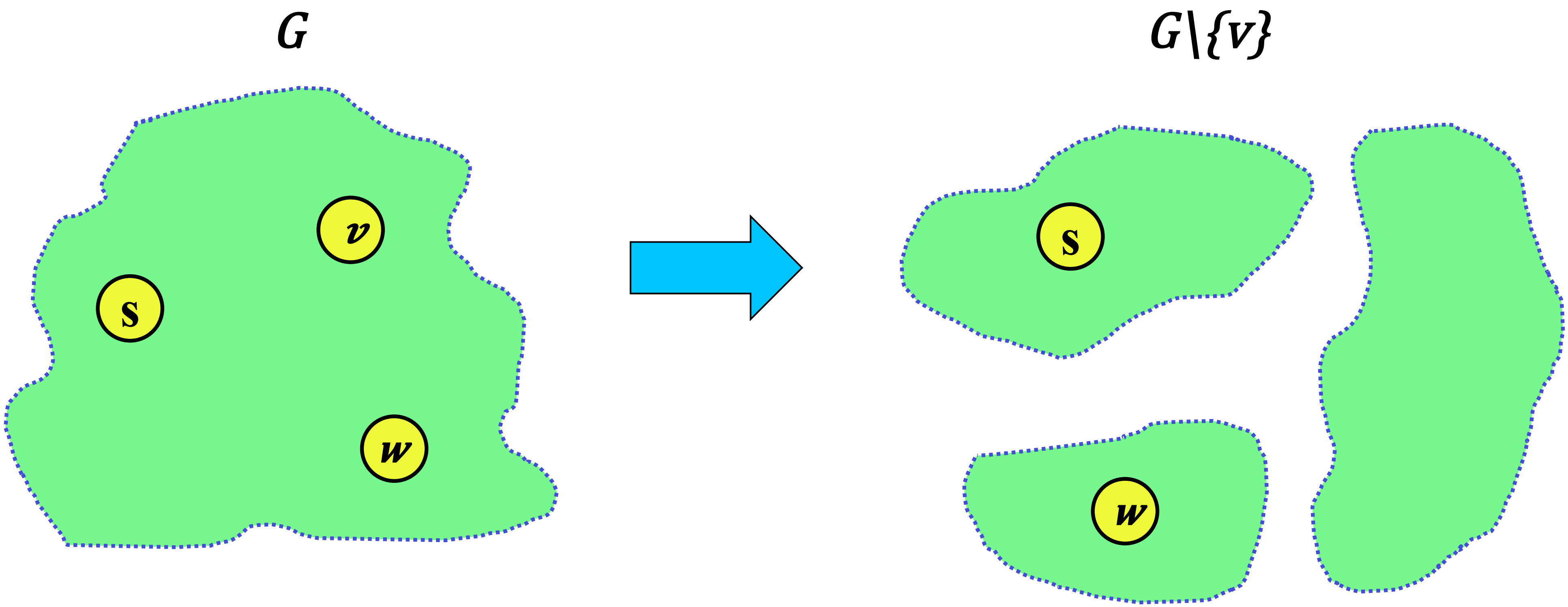
Proof: Let *v* be a strong articulation point



Strong Articulation Points

Theorem. *Let G = (V,E) be a strongly connected graph, and let $s \in V$ be any vertex in G. Then vertex $v \neq s$ is a strong articulation point in G if and only if $v \in D(s) \cup D^R(s)$.*

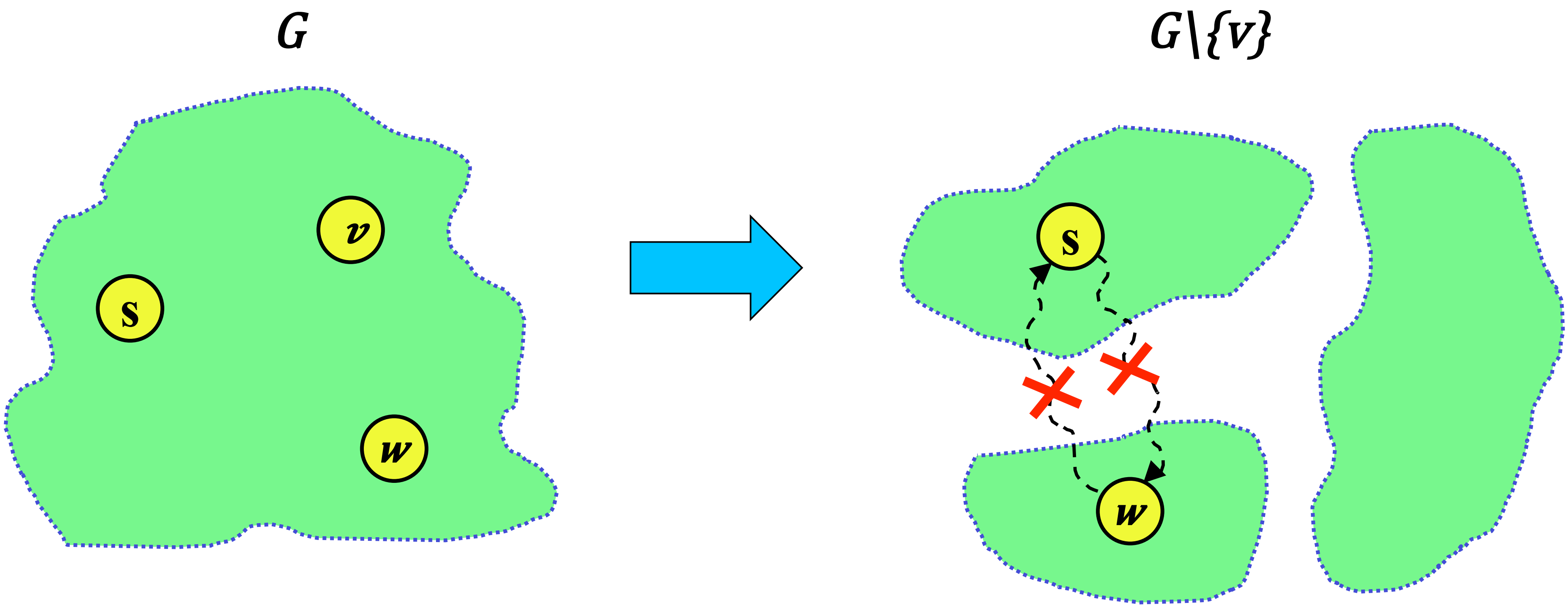
Proof: Let *v* be a strong articulation point



Strong Articulation Points

Theorem. *Let $G = (V,E)$ be a strongly connected graph, and let $s \in V$ be any vertex in G. Then vertex $v \neq s$ is a strong articulation point in G if and only if $v \in D(s) \cup D^R(s)$.*

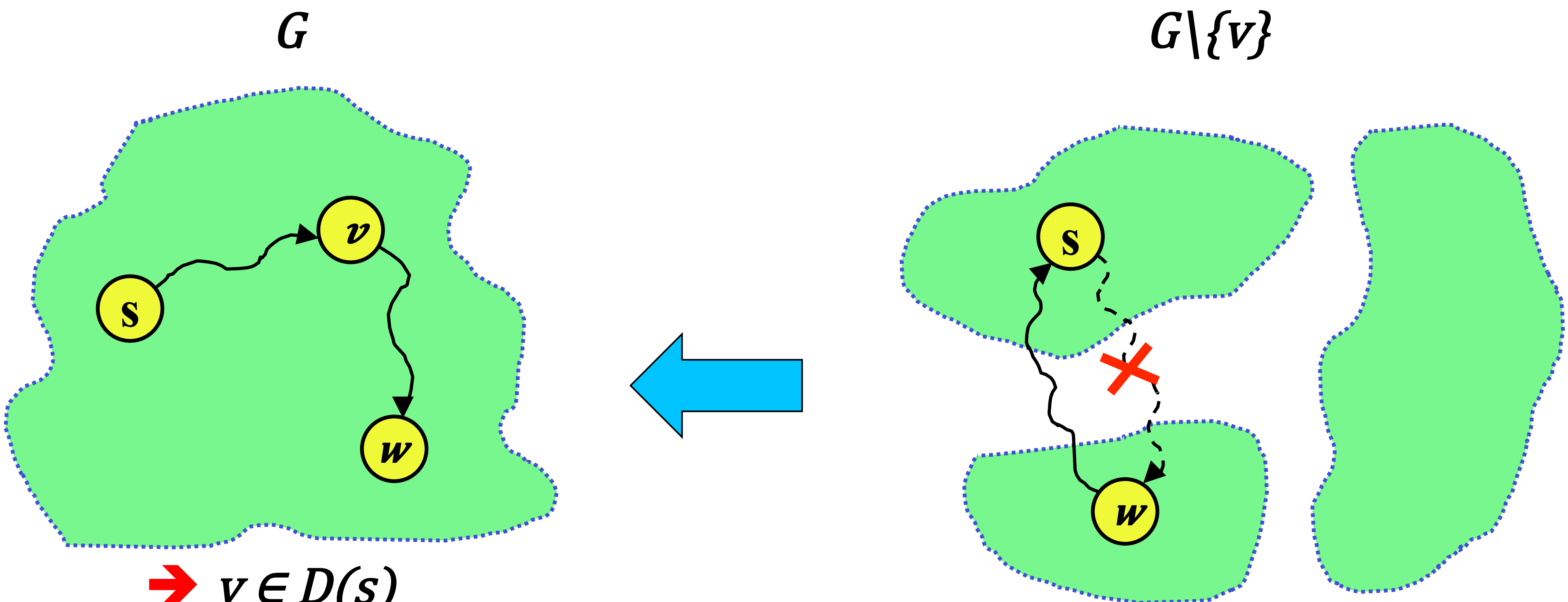
Proof: Let *v* be a strong articulation point



Strong Articulation Points

Theorem. *Let G = (V,E) be a strongly connected graph, and let s ∈ V be any vertex in G. Then vertex v ≠ s is a strong articulation point in G if and only if v ∈ D(s) ∪ D^R(s).*

Proof: Let *v* be a strong articulation point

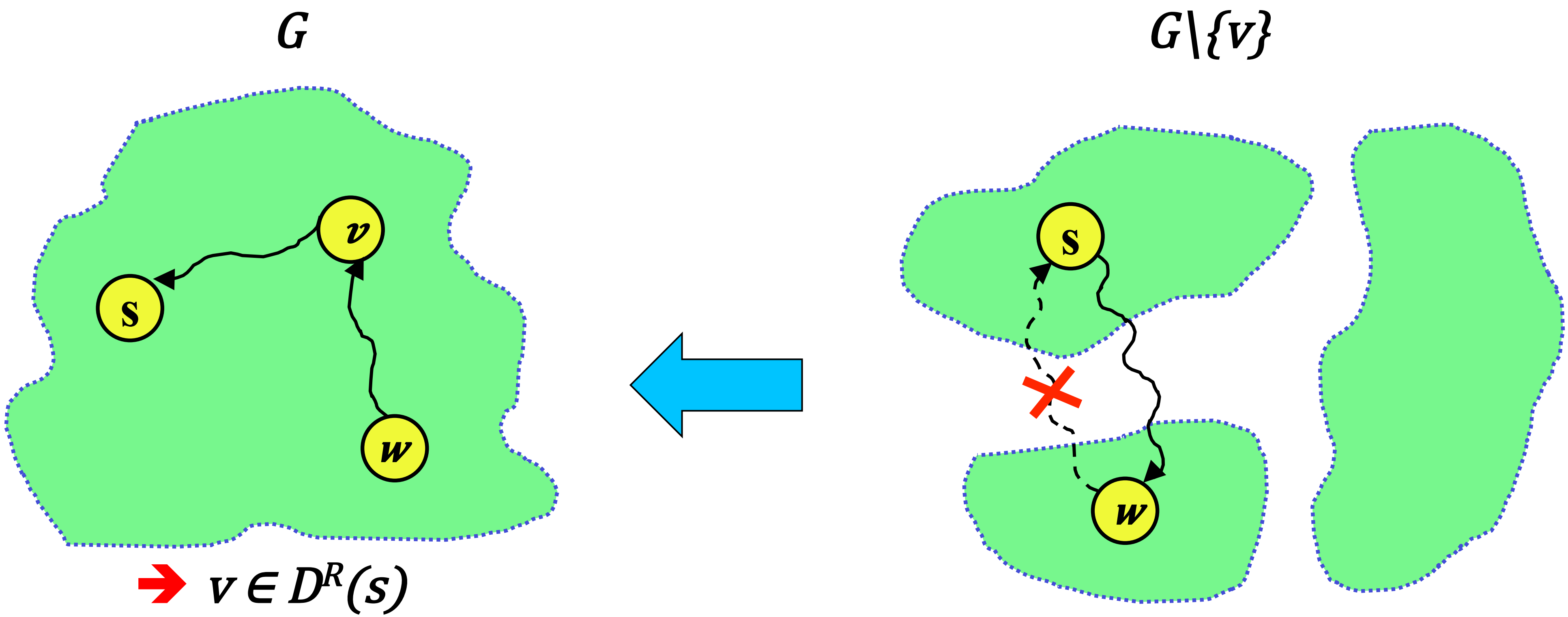


➔ $v \in D(s)$

Strong Articulation Points

Theorem. *Let G = (V,E) be a strongly connected graph, and let $s \in V$ be any vertex in G. Then vertex $v \neq s$ is a strong articulation point in G if and only if $v \in D(s) \cup D^R(s)$.*

Proof: Let *v* be a strong articulation point



Linear-Time Algorithm

Input: A strongly connected graph $G = (V, E)$, with n vertices and m edges.

Output: The strong articulation points of G.

1. Choose arbitrarily a vertex $s \in V$ in G, and test weather s is a strong articulation point in G. If s is an articulation point, output s.

2. Compute and output $D(s)$, the set of non-trivial dominators in the flowgraph $G(s) = (V,E,s)$.

3. Compute the reversal graph $G^R = (V,E^R)$.

4. Compute and output $D^R(s)$, the set of non-trivial dominators in the flowgraph $G^R(s) = (V,E^R,s)$.

Total time is $O(m+n)$

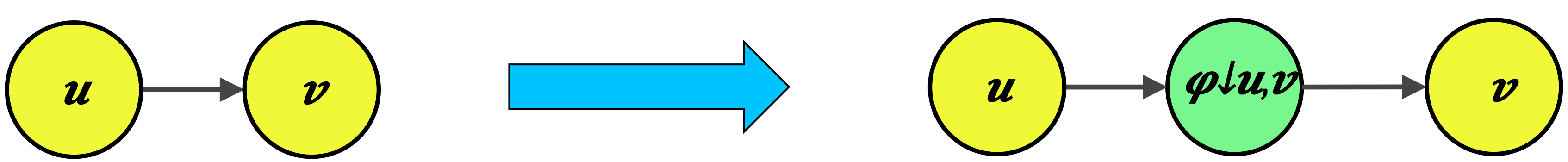
Strong Bridges

Strong Bridges

1. Reduction:

Lemma. *If there is an algorithm to compute the strong articulation points of a strongly connected graph in time T(m,n), then there is algorithm to compute the strong bridges of a strongly connected graph in time O(m + n + T(2m, n + m)).*

"Proof" :



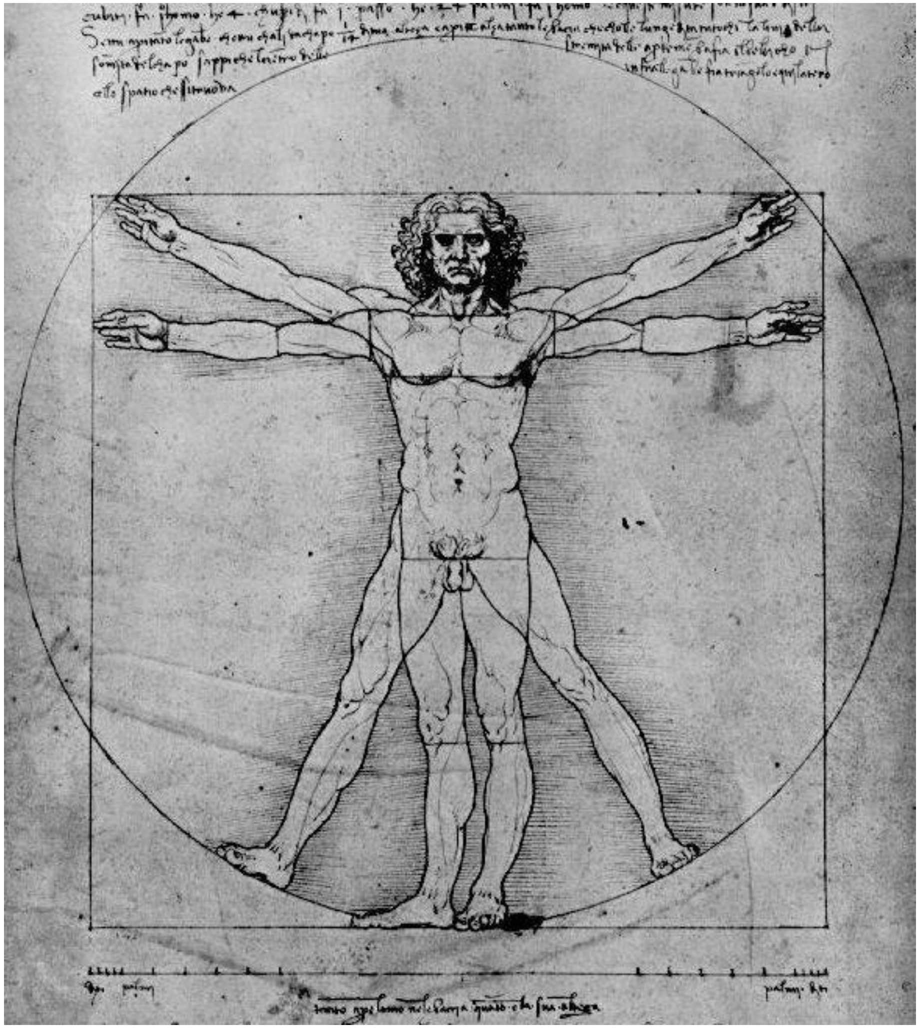
Mainly of theoretical interest (# vertices blows up)

SOFSEM 2012 :=

38th International Conference on Current Trends in Theory and Practice of Computer Science

January 21–27, 2012

Theory

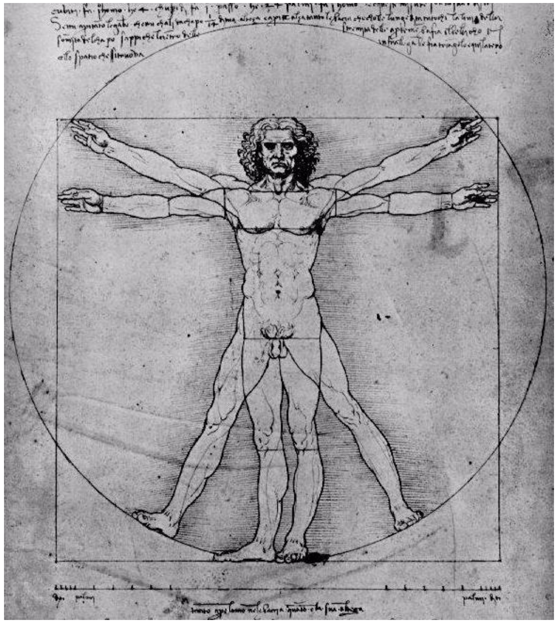


In theory, theory and practice are the same.

The real world out there…

In practice, theory and practice are different...

Bridging the Gap between Theory and Practice



Theory is when you know something, but it doesn't work.

Wish to combine theory and practice…

…i.e., nothing works and you don't know why.

Practice is when something works, but you don't know why.

Strong Bridges

2. Edge Dominators

Edge (u,v) is *dominator* of vertex w if every path from s to v contains edge (u,v)

If (u,v) is an edge dominator of vertex w, and every other edge dominator of u dominates w, we say that (u,v) is an *immediate edge dominator* of vertex w.

If a vertex has an edge dominator, then it has a *unique* immediate edge dominator.

With some care, able to extend all the theory from (vertex) dominators to edge dominators.

Given a flowgraph $G(s) = (V,E,s)$, edge dominators can be computed in time $O(m+n)$. Need to re-implement code for dominators.

Edge Dominators in Practice

Lemma. [Tarjan 1974] *Let $G = (V,E,s)$ be a flowgraph and let T be a DFS tree of G with start vertex s. Edge (v,w) is an edge dominator in flowgraph G if and only if all of the following conditions are met:*
- (v,w) is a tree edge,
- w has no entering forward edge or cross edge, and
- there is no back edge (x,w) such that w does not dominate x.

Need to (1) compute dominator tree $DT(s)$ and (2) check whether w ancestor of x in $DT(s)$ for back edge (x,w).

Given a flowgraph $G(s) = (V,E,s)$, edge dominators can be computed in time $O(m+n)$. Reuse code for (vertex) dominators. More efficient in practice. But still slightly slower than (vertex) dominators.

Computing All Strong Bridges

Given a strongly connected graph $G=(V,E)$, let

- $G(s) = (V,E,s)$ be the flowgraph with start vertex s
- $ED(s)$ the set of *edge dominators* in $G(s)$
- $G^R(s) = (V,E^R,s)$ be the flowgraph with start vertex s
- $ED^R(s)$ the set of *edge dominators* in $G^R(s)$

Theorem. *Let $G = (V,E)$ be a strongly connected graph, and let $s \in V$ be any vertex in G. Then edge (u,v) is a strong bridge in G if and only if $(u,v) \in ED(s) \cup ED^R(s)$.*

Incidentally, this proves also that can be at most $2n-2$ strong bridges in a directed graph.

Outline of the Talk

1. Preliminary Definitions
2. Algorithms for strong articulation points and strong bridges
3. **Preliminary Experiments on Large Scale Graphs**
4. Conclusions

Preliminary Experiments

CPU Intel Xeon X5650 (6 cores) @ 2.67GHz

12MB cache

32GB of DDR3 RAM @ 1GHz

Linux Red Hat 4.1.2-46 (Kernel 2.6.18)

Java Virtual Machine 1.6.0_16 (64-Bit)

WebGraph library 3.0.1

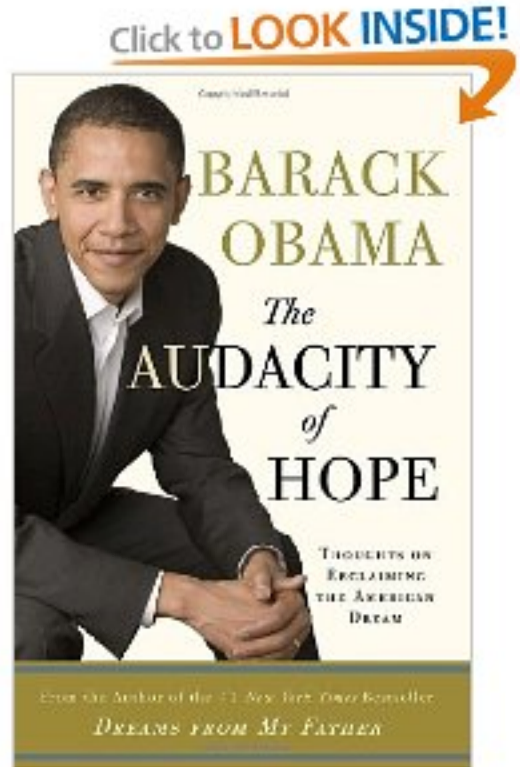
Implementations written in Java to exploit features offered by WebGraph (designed to deal with large graphs)

Datasets

Real-World Large Scale Graphs (billion edges):

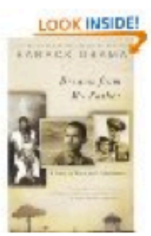
- Web Graphs
  (nodes webpages, edges hyperlinks)
- Social Graphs
  (social networks, edges represent interactions between people)
- Communication Graphs
  (email networks)
- Peer2Peer
  (nodes hosts in P2P network topology, edges connections between P2P hosts)
- Product Co-Purchase Graphs
  (nodes products, edges link commonly co-purchased products)

Product Co-Purchase



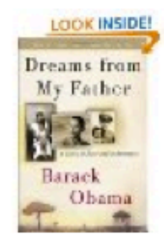
Customers Who Bought This Item Also Bought

Page 1 of 25



Dreams from My Father: A Story of Race and Inheri... by Barack Obama
(622)
$10.17

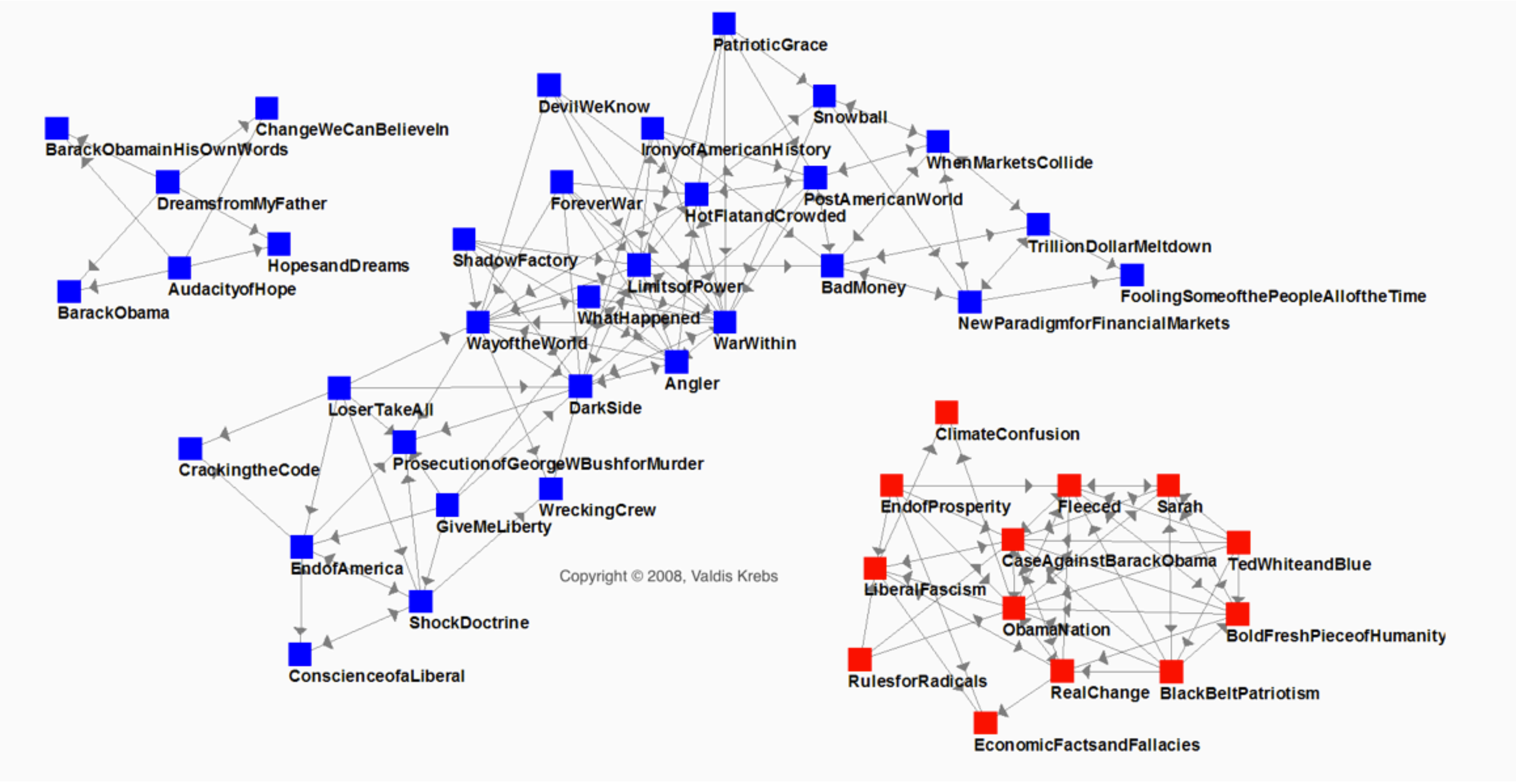
My Life by Bill Clinton
(708)
$22.63



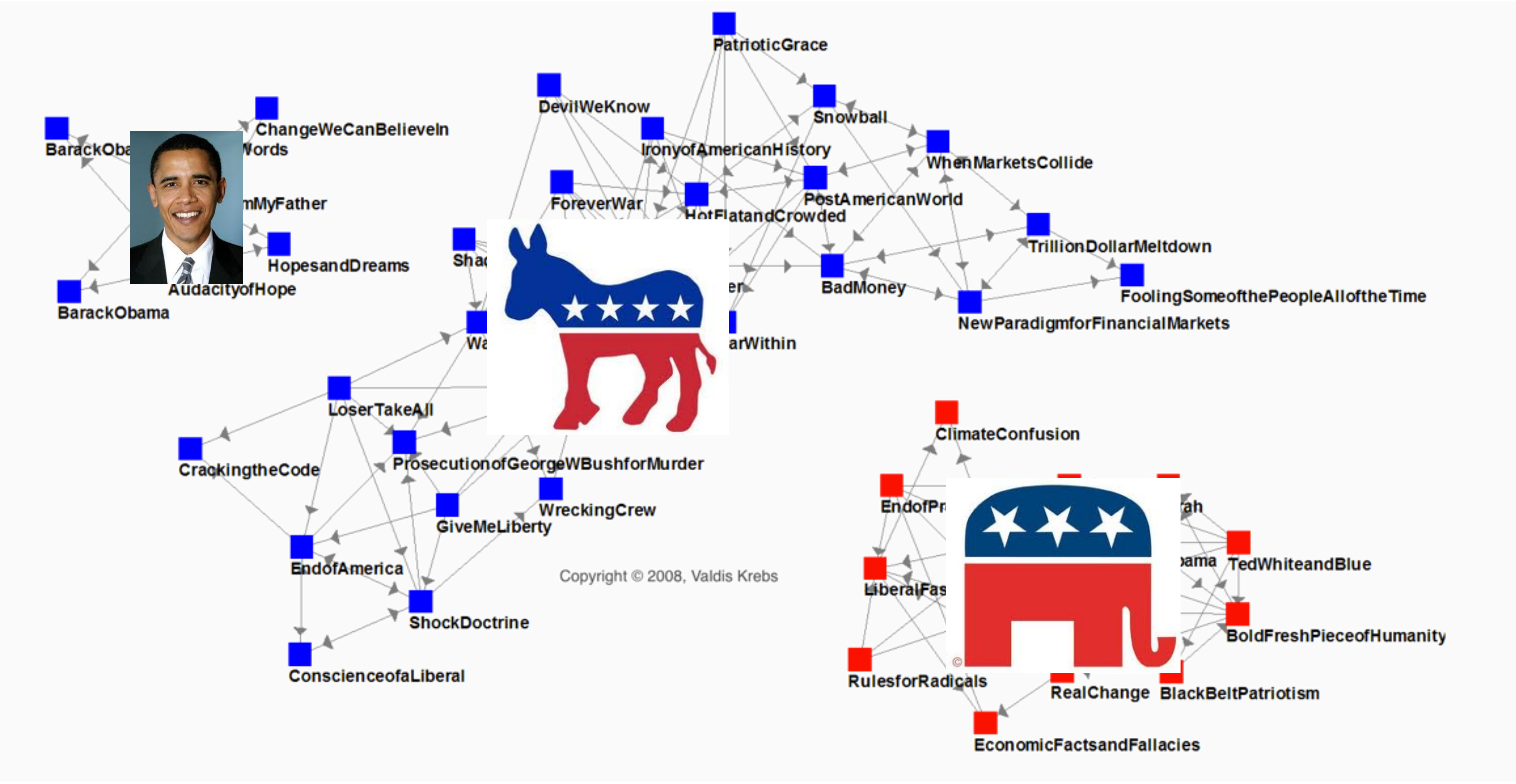
Dreams from My Father: A Story of Race and Inheri... by Barack Obama
(622)
$17.13

Change We Can Believe In: Barack Obama's Plan... by Barack Obama
(69)
$10.98

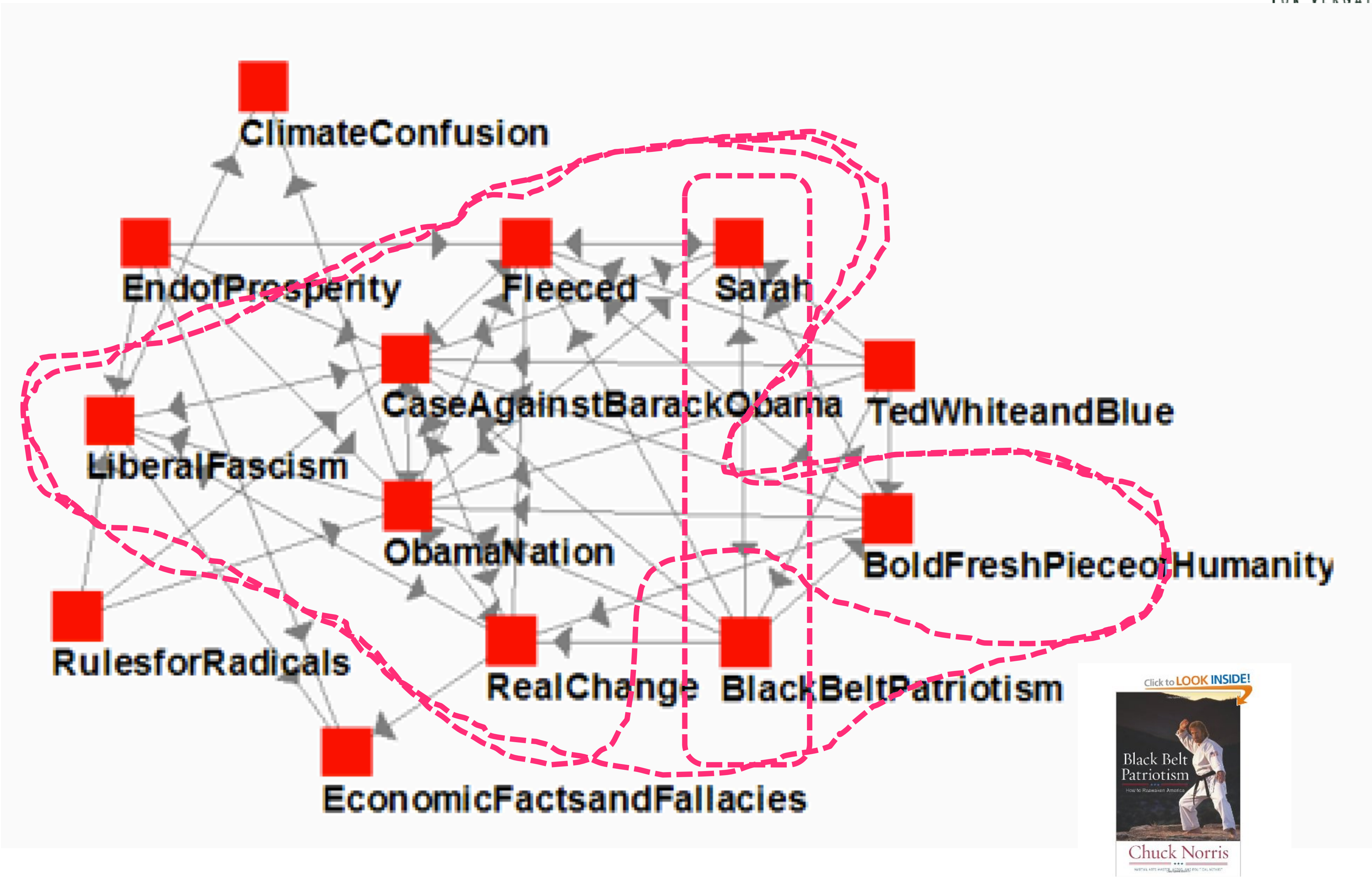
Product Co-Purchase



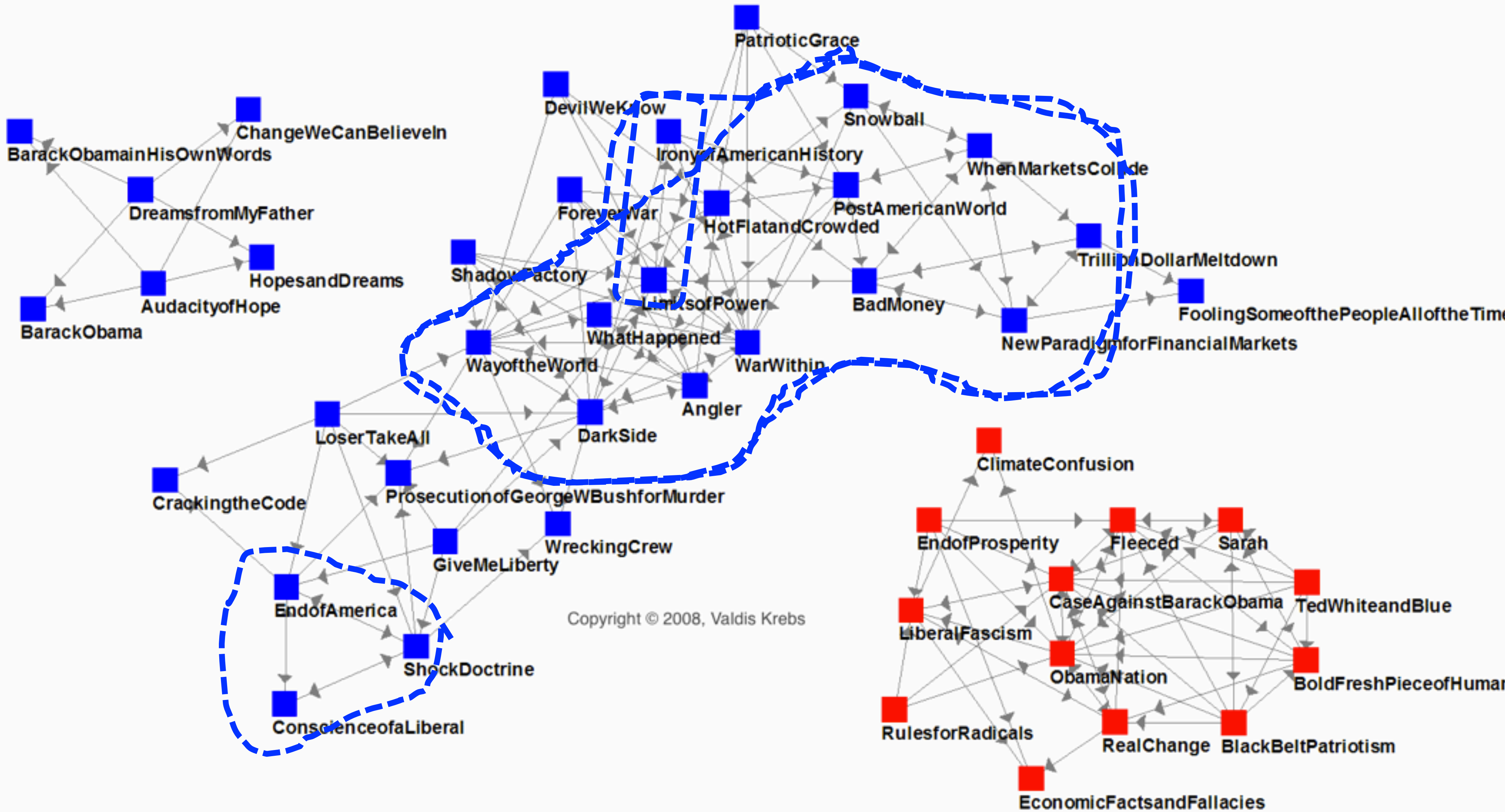
Product Co-Purchase



Product Co-Purchase



Product Co-Purchase



Datasets

Graph	Type	Repository	n	m	δ_{avg}
p2p-Gnutella04	Peer2peer	SNAP	11 K	40 K	3.7
wiki-Vote	Social	SNAP	7 K	103 K	14.5
enron	Communication	WebGraph	69 K	276 K	4.0
email-EuAll	Communication	SNAP	265 K	420 K	1.58
soc-Epinions1	Social	SNAP	76 K	509 K	30.5
soc-Slashdot0811	Social	SNAP	77 K	905 K	11.7
soc-Slashdot0902	Social	SNAP	82 K	948 K	11.5
amazon0302	Product co-purchase	SNAP	262 K	1.2 M	4.7
web-NotreDame	Web	WebGraph	325 K	1.4M	4.6
uk-2007-05@100K	Web	WebGraph	100 K	3 M	30.5
cnr-2000	Web	WebGraph	325 K	3.2 M	9.8
amazon0312	Product co-purchase	SNAP	400 K	3.2 M	8.0
amazon-2008	Product co-purchase	SNAP	735 K	5.1 M	7.0
wiki-Talk	Communication	SNAP	2.3 M	5.0 M	2.1
web-Google	Web	SNAP	875 K	5.1 M	5.8
web-BerkStan	Web	SNAP	685 K	7.6 M	11.0
in-2004	Web	WebGraph	1.3 M	16.9 M	12.2
eu-2005	Web	WebGraph	862 K	19.2 M	22.3
uk-2007-05@1M	Web	WebGraph	1 M	41.2 M	41.2
soc-LiveJournal1	Social	SNAP	4.8 M	68.9 M	14.2
ljournal-2008	Social	SNAP	5.3 M	79.0 M	14.7
indochina-2004	Web	WebGraph	7.4 M	194 M	26.1
uk-2002	Web	WebGraph	18.5M	298 M	16.1
arabic-2005	Web	WebGraph	22.7 M	640 M	28.1

Analysis of SAPs and SBs

Graph	n	m	$\#sap$	$\#sb$	n_{scc}	m_{scc}	$\#sap_{scc}$	$\#sb_{scc}$	ILS(SAP)	ILS(SB)
p2p-Gnutella04	11 K	40 K	1.3 K	1.6 K	4.3 K	18.7 K	1.3 K	1.6 K	0.15	0.18
wiki-Vote	7 K	103 K	143	152	1.3 K	39.4 K	143	152	0.18	0.18
enron	69 K	276 K	796	4.9 K	8.2 K	147 K	781	4.8 K	0.58	0.69
email-EuAll	265 K	420 K	962	46.0 K	34 K	151 K	960	46.0 K	1.32	1.80
soc-Epinions1	76 K	509 K	8.4 K	23.5 K	32 K	443 K	8.1 K	20.9 K	1.07	1.20
soc-Slashdot0811	77 K	905 K	14.0 K	417	70 K	888 K	13.0 K	3	1.76	2.19
soc-Slashdot0902	82 K	948 K	14.2 K	501	71 K	912 K	14.1 K	69	1.79	2.35
amazon0302	262 K	1.2 M	71.9 K	75.7 K	241 K	11.3 M	69.6 K	73.3 K	3.55	4.77
web-NotreDame	325 K	1.4 M	13.8 K	61.8 K	54 K	304 K	9.6 K	31.9 K	2.48	3.27
uk-2007-05@100K	100 K	3 M	9.4 K	47.0 K	53 K	1.6 M	2.8 K	16.8 K	3.00	3.46
cnr-2000	325 K	3.2 M	32.5 K	104 K	112 K	1.6 M	14.6 K	44.1 K	4.28	5.08
amazon0312	400 K	3.2 M	69.5 K	83.2 K	380 K	3.0 M	69.0 K	82.6 K	11.37	12.40
amazon-2008	735 K	5.1 M	103 K	159 K	627 K	4.7 M	102 K	156 K	25.81	21.89
wiki-Talk	2.3 M	5.0 M	14.8 K	86.7 K	111 K	14.7 M	14.8 K	85.5 K	19.02	18.58
web-Google	875 K	5.1 M	102 K	267 K	434 K	3.4 M	89.8 K	211 K	13.59	15.48
web-BerkStan	685 K	7.6 M	108 K	297 K	334 K	4.5 M	53.6 K	164 K	9.91	12.15
in-2004	1.3 M	16.9 M	82.0 K	421 K	480 K	7.8 M	33.5 K	216 K	32.39	39.02
eu-2005	862 K	19.2 M	104 K	160 K	752 K	17.9 M	99.3 K	146 K	23.95	27.67
uk-2007-05@1M	1 M	41.2 M	147 K	415 K	593 K	22.0 M	82.5 K	259 K	20.90	24.51
soc-LiveJournal1	4.8 M	68.9 M	654 K	1.3 M	3.8 M	65.8 M	649 K	1.3 M	260.03	273.60
ljournal-2008	5.3 M	79.0 M	734 K	1.3 M	4.1 M	74.9 M	727 K	1.3 M	275.53	299.57
indochina-2004	7.4 M	194 M	774 K	2.2 M	3.8 M	98.8 M	503 K	1.4 M	155.83	192.06
uk-2002	18.5 M	298 M	2.3 M	6.1 M	12.0 M	232 M	1.8 M	4.8 M	404.92	478.13
arabic-2005	22.7 M	640 M	2.7 M	6.7 M	15.1 M	473 M	2.2 M	5.2 M	681.47	837.89

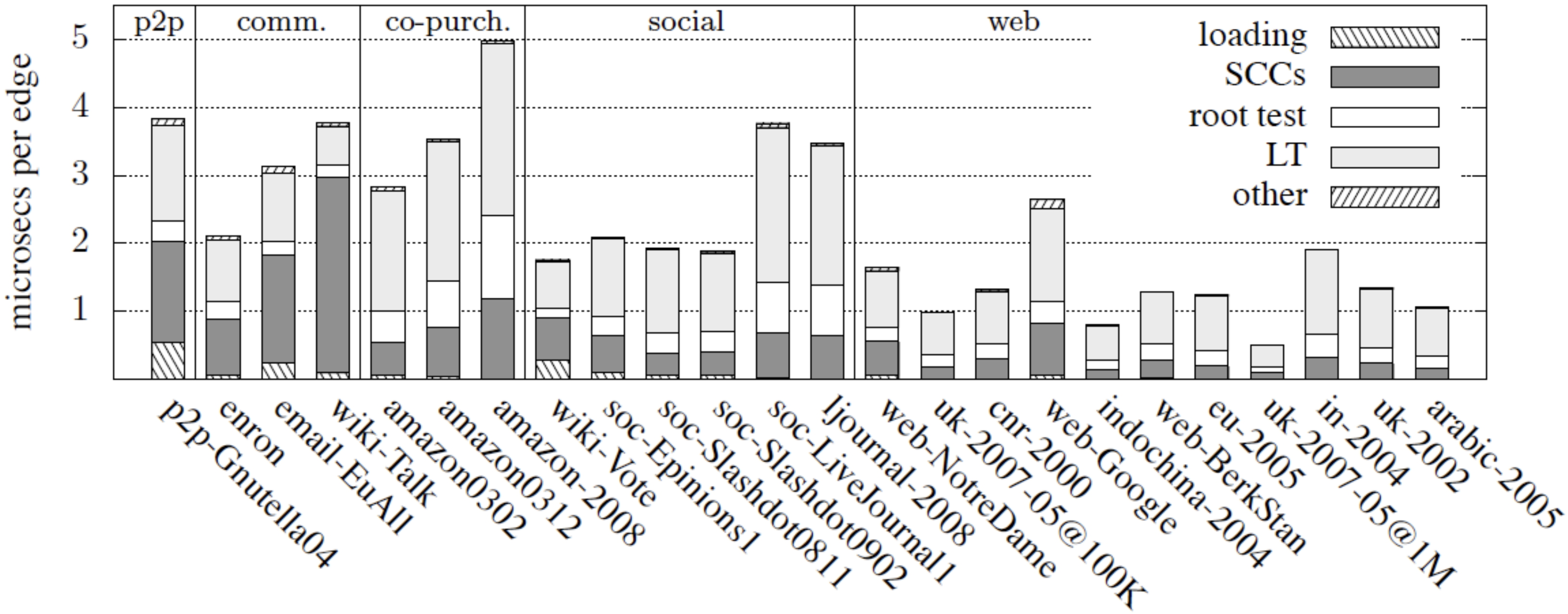
Running times (secs)

Analysis of SAPs and SBs

Graph	n	m	$\#sap$	$\#sb$	n_{scc}	m_{scc}	$\#sap_{scc}$	$\#sb_{scc}$	ILS(SAP)	ILS(SB)
p2p-Gnutella04	11 K	40 K	1.3 K	1.6 K	4.3 K	18.7 K	1.3 K	1.6 K	0.15	0.18
wiki-Vote	7 K	103 K	143	152	1.3 K	39.4 K	143	152	0.18	0.18
enron	69 K	276 K	796	4.9 K	8.2 K	147 K	781	4.8 K	0.58	0.69
email-EuAll	265 K	420 K	962	46.0 K	34 K	151 K	960	46.0 K	1.32	1.80
soc-Epinions1	76 K	509 K	8.4 K	23.5 K	32 K	443 K	8.1 K	20.9 K	1.07	1.20
soc-Slashdot0811	77 K	905 K	14.0 K	417	70 K	888 K	13.0 K	3	1.76	2.19
soc-Slashdot0902	82 K	948 K	14.2 K	501	71 K	912 K	14.1 K	69	1.79	2.35
amazon0302	262 K	1.2 M	71.9 K	75.7 K	241 K	11.3 M	69.6 K	73.3 K	3.55	4.77
web-NotreDame	325 K	1.4 M	13.8 K	61.8 K	54 K	304 K	9.6 K	31.9 K	2.48	3.27
uk-2007-05@100K	100 K	3 M	9.4 K	47.0 K	53 K	1.6 M	2.8 K	16.8 K	3.00	3.46
cnr-2000	325 K	3.2 M	32.5 K	104 K	112 K	1.6 M	14.6 K	44.1 K	4.28	5.08
amazon0312	400 K	3.2 M	69.5 K	83.2 K	380 K	3.0 M	69.0 K	82.6 K	11.37	12.40
amazon-2008	735 K	5.1 M	103 K	159 K	627 K	4.7 M	102 K	156 K	25.81	21.89
wiki-Talk	2.3 M	5.0 M	14.8 K	86.7 K	111 K	14.7 M	14.8 K	85.5 K	19.02	18.58
web-Google	875 K	5.1 M	102 K	267 K	434 K	3.4 M	89.8 K	211 K	13.59	15.48
web-BerkStan	685 K	7.6 M	108 K	297 K	334 K	4.5 M	53.6 K	164 K	9.91	12.15
in-2004	1.3 M	16.9 M	82.0 K	421 K	480 K	7.8 M	33.5 K	216 K	32.39	39.02
eu-2005	862 K	19.2 M	104 K	160 K	752 K	17.9 M	99.3 K	146 K	23.95	27.67
uk-2007-05@1M	1 M	41.2 M	147 K	415 K	593 K	22.0 M	82.5 K	259 K	20.90	24.51
soc-LiveJournal1	4.8 M	68.9 M	654 K	1.3 M	3.8 M	65.8 M	649 K	1.3 M	260.03	273.60
ljournal-2008	5.3 M	79.0 M	734 K	1.3 M	4.1 M	74.9 M	727 K	1.3 M	275.53	299.57
indochina-2004	7.4 M	194 M	774 K	2.2 M	3.8 M	98.8 M	503 K	1.4 M	155.83	192.06
uk-2002	18.5 M	298 M	2.3 M	6.1 M	12.0 M	232 M	1.8 M	4.8 M	404.92	478.13
arabic-2005	22.7 M	640 M	2.7 M	6.7 M	15.1 M	473 M	2.2 M	5.2 M	681.47	837.89

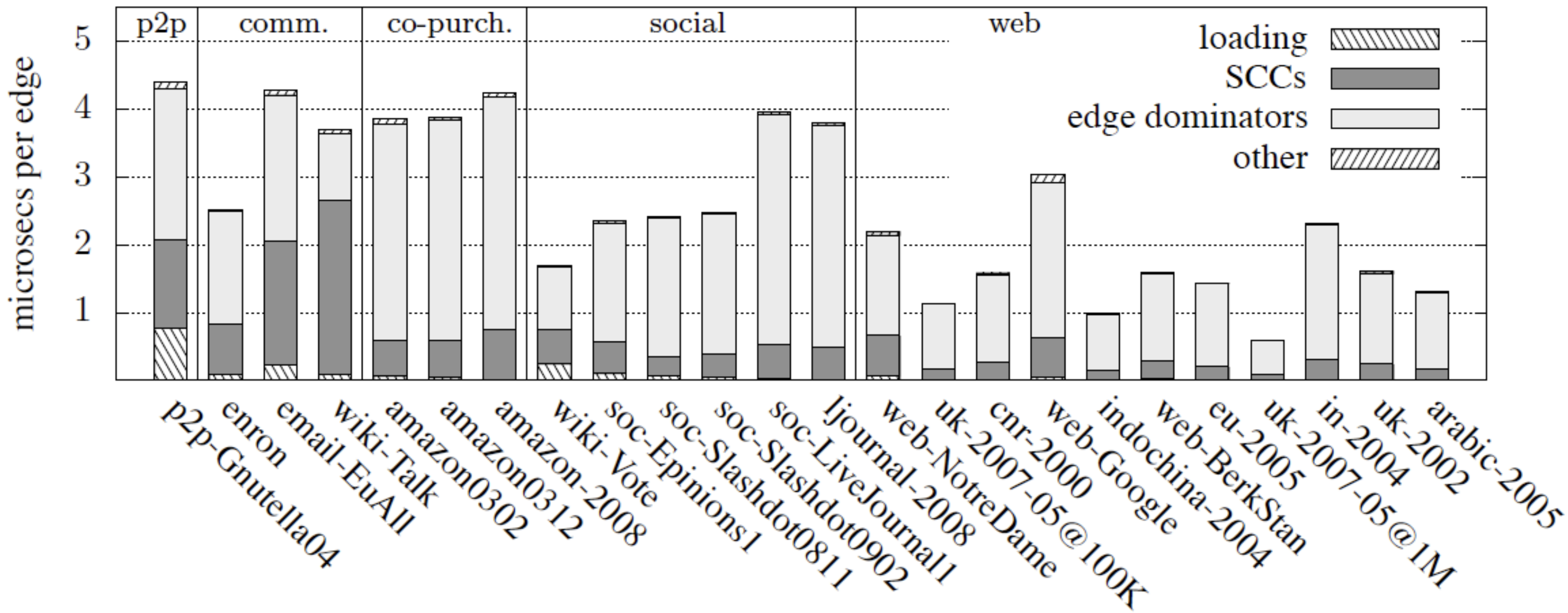
…able to process massive graphs (billion edges) in 10-15 minutes

Faster Implementations?



SAP

SB



Analysis of SAPs and SBs

Graph	n	m	$\#sap$	$\#sb$	n_{scc}	m_{scc}	$\#sap_{scc}$	$\#sb_{scc}$	ILS(SAP)	ILS(SB)
p2p-Gnutella04	11 K	40 K	1.3 K	1.6 K	4.3 K	18.7 K	1.3 K	1.6 K	0.15	0.18
wiki-Vote	7 K	103 K	143	152	1.3 K	39.4 K	143	152	0.18	0.18
enron	69 K	276 K	796	4.9 K	8.2 K	147 K	781	4.8 K	0.58	0.69
email-EuAll	265 K	420 K	962	46.0 K	34 K	151 K	960	46.0 K	1.32	1.80
soc-Epinions1	76 K	509 K	8.4 K	23.5 K	32 K	443 K	8.1 K	20.9 K	1.07	1.20
soc-Slashdot0811	77 K	905 K	14.0 K	417	70 K	888 K	13.0 K	3	1.76	2.19
soc-Slashdot0902	82 K	948 K	14.2 K	501	71 K	912 K	14.1 K	69	1.79	2.35
amazon0302	262 K	1.2 M	71.9 K	75.7 K	241 K	11.3 M	69.6 K	73.3 K	3.55	4.77
web-NotreDame	325 K	1.4 M	13.8 K	61.8 K	54 K	304 K	9.6 K	31.9 K	2.48	3.27
uk-2007-05@100K	100 K	3 M	9.4 K	47.0 K	53 K	1.6 M	2.8 K	16.8 K	3.00	3.46
cnr-2000	325 K	3.2 M	32.5 K	104 K	112 K	1.6 M	14.6 K	44.1 K	4.28	5.08
amazon0312	400 K	3.2 M	69.5 K	83.2 K	380 K	3.0 M	69.0 K	82.6 K	11.37	12.40
amazon-2008	735 K	5.1 M	103 K	159 K	627 K	4.7 M	102 K	156 K	25.81	21.89
wiki-Talk	2.3 M	5.0 M	14.8 K	86.7 K	111 K	14.7 M	14.8 K	85.5 K	19.02	18.58
web-Google	875 K	5.1 M	102 K	267 K	434 K	3.4 M	89.8 K	211 K	13.59	15.48
web-BerkStan	685 K	7.6 M	108 K	297 K	334 K	4.5 M	53.6 K	164 K	9.91	12.15
in-2004	1.3 M	16.9 M	82.0 K	421 K	480 K	7.8 M	33.5 K	216 K	32.39	39.02
eu-2005	862 K	19.2 M	104 K	160 K	752 K	17.9 M	99.3 K	146 K	23.95	27.67
uk-2007-05@1M	1 M	41.2 M	147 K	415 K	593 K	22.0 M	82.5 K	259 K	20.90	24.51
soc-LiveJournal1	4.8 M	68.9 M	654 K	1.3 M	3.8 M	65.8 M	649 K	1.3 M	260.03	273.60
ljournal-2008	5.3 M	79.0 M	734 K	1.3 M	4.1 M	74.9 M	727 K	1.3 M	275.53	299.57
indochina-2004	7.4 M	194 M	774 K	2.2 M	3.8 M	98.8 M	503 K	1.4 M	155.83	192.06
uk-2002	18.5 M	298 M	2.3 M	6.1 M	12.0 M	232 M	1.8 M	4.8 M	404.92	478.13
arabic-2005	22.7 M	640 M	2.7 M	6.7 M	15.1 M	473 M	2.2 M	5.2 M	681.47	837.89

SAPs appear often (15-25% co-purchase, 11-18% social)

Analysis of SAPs and SBs

Graph	n	m	$\#sap$	$\#sb$	n_{scc}	m_{scc}	$\#sap_{scc}$	$\#sb_{scc}$	ILS(SAP)	ILS(SB)
p2p-Gnutella04	11 K	40 K	1.3 K	1.6 K	4.3 K	18.7 K	1.3 K	1.6 K	0.15	0.18
wiki-Vote	7 K	103 K	143	152	1.3 K	39.4 K	143	152	0.18	0.18
enron	69 K	276 K	796	4.9 K	8.2 K	147 K	781	4.8 K	0.58	0.69
email-EuAll	265 K	420 K	962	46.0 K	34 K	151 K	960	46.0 K	1.32	1.80
soc-Epinions1	76 K	509 K	8.4 K	23.5 K	32 K	443 K	8.1 K	20.9 K	1.07	1.20
soc-Slashdot0811	77 K	905 K	14.0 K	417	70 K	888 K	13.0 K	3	1.76	2.19
soc-Slashdot0902	82 K	948 K	14.2 K	501	71 K	912 K	14.1 K	69	1.79	2.35
amazon0302	262 K	1.2 M	71.9 K	75.7 K	241 K	11.3 M	69.6 K	73.3 K	3.55	4.77
web-NotreDame	325 K	1.4 M	13.8 K	61.8 K	54 K	304 K	9.6 K	31.9 K	2.48	3.27
uk-2007-05@100K	100 K	3 M	9.4 K	47.0 K	53 K	1.6 M	2.8 K	16.8 K	3.00	3.46
cnr-2000	325 K	3.2 M	32.5 K	104 K	112 K	1.6 M	14.6 K	44.1 K	4.28	5.08
amazon0312	400 K	3.2 M	69.5 K	83.2 K	380 K	3.0 M	69.0 K	82.6 K	11.37	12.40
amazon-2008	735 K	5.1 M	103 K	159 K	627 K	4.7 M	102 K	156 K	25.81	21.89
wiki-Talk	2.3 M	5.0 M	14.8 K	86.7 K	111 K	14.7 M	14.8 K	85.5 K	19.02	18.58
web-Google	875 K	5.1 M	102 K	267 K	434 K	3.4 M	89.8 K	211 K	13.59	15.48
web-BerkStan	685 K	7.6 M	108 K	297 K	334 K	4.5 M	53.6 K	164 K	9.91	12.15
in-2004	1.3 M	16.9 M	82.0 K	421 K	480 K	7.8 M	33.5 K	216 K	32.39	39.02
eu-2005	862 K	19.2 M	104 K	160 K	752 K	17.9 M	99.3 K	146 K	23.95	27.67
uk-2007-05@1M	1 M	41.2 M	147 K	415 K	593 K	22.0 M	82.5 K	259 K	20.90	24.51
soc-LiveJournal1	4.8 M	68.9 M	654 K	1.3 M	3.8 M	65.8 M	649 K	1.3 M	260.03	273.60
ljournal-2008	5.3 M	79.0 M	734 K	1.3 M	4.1 M	74.9 M	727 K	1.3 M	275.53	299.57
indochina-2004	7.4 M	194 M	774 K	2.2 M	3.8 M	98.8 M	503 K	1.4 M	155.83	192.06
uk-2002	18.5 M	298 M	2.3 M	6.1 M	12.0 M	232 M	1.8 M	4.8 M	404.92	478.13
arabic-2005	22.7 M	640 M	2.7 M	6.7 M	15.1 M	473 M	2.2 M	5.2 M	681.47	837.89

The vast majority of SAPs are in big SCC (less for Web graphs)

Analysis of SAPs and SBs

Graph	n	m	$\#sap$	$\#sb$	n_{scc}	m_{scc}	$\#sap_{scc}$	$\#sb_{scc}$	ILS(SAP)	ILS(SB)
p2p-Gnutella04	11 K	40 K	1.3 K	1.6 K	4.3 K	18.7 K	1.3 K	1.6 K	0.15	0.18
wiki-Vote	7 K	103 K	143	152	1.3 K	39.4 K	143	152	0.18	0.18
enron	69 K	276 K	796	4.9 K	8.2 K	147 K	781	4.8 K	0.58	0.69
email-EuAll	265 K	420 K	962	46.0 K	34 K	151 K	960	46.0 K	1.32	1.80
soc-Epinions1	76 K	509 K	8.4 K	23.5 K	32 K	443 K	8.1 K	20.9 K	1.07	1.20
soc-Slashdot0811	77 K	905 K	14.0 K	417	70 K	888 K	13.0 K	3	1.76	2.19
soc-Slashdot0902	82 K	948 K	14.2 K	501	71 K	912 K	14.1 K	69	1.79	2.35
amazon0302	262 K	1.2 M	71.9 K	75.7 K	241 K	11.3 M	69.6 K	73.3 K	3.55	4.77
web-NotreDame	325 K	1.4 M	13.8 K	61.8 K	54 K	304 K	9.6 K	31.9 K	2.48	3.27
uk-2007-05@100K	100 K	3 M	9.4 K	47.0 K	53 K	1.6 M	2.8 K	16.8 K	3.00	3.46
cnr-2000	325 K	3.2 M	32.5 K	104 K	112 K	1.6 M	14.6 K	44.1 K	4.28	5.08
amazon0312	400 K	3.2 M	69.5 K	83.2 K	380 K	3.0 M	69.0 K	82.6 K	11.37	12.40
amazon-2008	735 K	5.1 M	103 K	159 K	627 K	4.7 M	102 K	156 K	25.81	21.89
wiki-Talk	2.3 M	5.0 M	14.8 K	86.7 K	111 K	14.7 M	14.8 K	85.5 K	19.02	18.58
web-Google	875 K	5.1 M	102 K	267 K	434 K	3.4 M	89.8 K	211 K	13.59	15.48
web-BerkStan	685 K	7.6 M	108 K	297 K	334 K	4.5 M	53.6 K	164 K	9.91	12.15
in-2004	1.3 M	16.9 M	82.0 K	421 K	480 K	7.8 M	33.5 K	216 K	32.39	39.02
eu-2005	862 K	19.2 M	104 K	160 K	752 K	17.9 M	99.3 K	146 K	23.95	27.67
uk-2007-05@1M	1 M	41.2 M	147 K	415 K	593 K	22.0 M	82.5 K	259 K	20.90	24.51
soc-LiveJournal1	4.8 M	68.9 M	654 K	1.3 M	3.8 M	65.8 M	649 K	1.3 M	260.03	273.60
ljournal-2008	5.3 M	79.0 M	734 K	1.3 M	4.1 M	74.9 M	727 K	1.3 M	275.53	299.57
indochina-2004	7.4 M	194 M	774 K	2.2 M	3.8 M	98.8 M	503 K	1.4 M	155.83	192.06
uk-2002	18.5 M	298 M	2.3 M	6.1 M	12.0 M	232 M	1.8 M	4.8 M	404.92	478.13
arabic-2005	22.7 M	640 M	2.7 M	6.7 M	15.1 M	473 M	2.2 M	5.2 M	681.47	837.89

SBs are less frequent (except for email graphs)

Analysis of SAPs and SBs

Graph	n	m	$\#sap$	$\#sb$	n_{scc}	m_{scc}	$\#sap_{scc}$	$\#sb_{scc}$	ILS(SAP)	ILS(SB)
p2p-Gnutella04	11 K	40 K	1.3 K	1.6 K	4.3 K	18.7 K	1.3 K	1.6 K	0.15	0.18
wiki-Vote	7 K	103 K	143	152	1.3 K	39.4 K	143	152	0.18	0.18
enron	69 K	276 K	796	4.9 K	8.2 K	147 K	781	4.8 K	0.58	0.69
email-EuAll	265 K	420 K	962	46.0 K	34 K	151 K	960	46.0 K	1.32	1.80
soc-Epinions1	76 K	509 K	8.4 K	23.5 K	32 K	443 K	8.1 K	20.9 K	1.07	1.20
soc-Slashdot0811	77 K	905 K	14.0 K	417	70 K	888 K	13.0 K	3	1.76	2.19
soc-Slashdot0902	82 K	948 K	14.2 K	501	71 K	912 K	14.1 K	69	1.79	2.35
amazon0302	262 K	1.2 M	71.9 K	75.7 K	241 K	11.3 M	69.6 K	73.3 K	3.55	4.77
web-NotreDame	325 K	1.4 M	13.8 K	61.8 K	54 K	304 K	9.6 K	31.9 K	2.48	3.27
uk-2007-05@100K	100 K	3 M	9.4 K	47.0 K	53 K	1.6 M	2.8 K	16.8 K	3.00	3.46
cnr-2000	325 K	3.2 M	32.5 K	104 K	112 K	1.6 M	14.6 K	44.1 K	4.28	5.08
amazon0312	400 K	3.2 M	69.5 K	83.2 K	380 K	3.0 M	69.0 K	82.6 K	11.37	12.40
amazon-2008	735 K	5.1 M	103 K	159 K	627 K	4.7 M	102 K	156 K	25.81	21.89
wiki-Talk	2.3 M	5.0 M	14.8 K	86.7 K	111 K	14.7 M	14.8 K	85.5 K	19.02	18.58
web-Google	875 K	5.1 M	102 K	267 K	434 K	3.4 M	89.8 K	211 K	13.59	15.48
web-BerkStan	685 K	7.6 M	108 K	297 K	334 K	4.5 M	53.6 K	164 K	9.91	12.15
in-2004	1.3 M	16.9 M	82.0 K	421 K	480 K	7.8 M	33.5 K	216 K	32.39	39.02
eu-2005	862 K	19.2 M	104 K	160 K	752 K	17.9 M	99.3 K	146 K	23.95	27.67
uk-2007-05@1M	1 M	41.2 M	147 K	415 K	593 K	22.0 M	82.5 K	259 K	20.90	24.51
soc-LiveJournal1	4.8 M	68.9 M	654 K	1.3 M	3.8 M	65.8 M	649 K	1.3 M	260.03	273.60
ljournal-2008	5.3 M	79.0 M	734 K	1.3 M	4.1 M	74.9 M	727 K	1.3 M	275.53	299.57
indochina-2004	7.4 M	194 M	774 K	2.2 M	3.8 M	98.8 M	503 K	1.4 M	155.83	192.06
uk-2002	18.5 M	298 M	2.3 M	6.1 M	12.0 M	232 M	1.8 M	4.8 M	404.92	478.13
arabic-2005	22.7 M	640 M	2.7 M	6.7 M	15.1 M	473 M	2.2 M	5.2 M	681.47	837.89

SBs are also in SCC (less for social graphs)

Some Properties of SAPs

Graph	δ^-_{avg} V	δ^-_{avg} sap	δ^+_{avg} V	δ^+_{avg} sap	PR_{avg} V	PR_{avg} sap
p2p-Gnutella04	3.68	4.87	3.68	9.60	$9.19 \cdot 10^{-5}$	$1.12 \cdot 10^{-4}$
enron	3.99	62.79	3.99	103.48	$1.46 \cdot 10^{-5}$	$1.63 \cdot 10^{-4}$
email-EuAll	1.58	280.43	1.58	103.06	$3.80 \cdot 10^{-6}$	$4.29 \cdot 10^{-4}$
wiki-Talk	2.1	69.35	2.10	290.50	$4.18 \cdot 10^{-7}$	$2.86 \cdot 10^{-6}$
amazon0302	4.71	4.65	4.71	4.89	$3.82 \cdot 10^{-6}$	$3.42 \cdot 10^{-6}$
amazon0312	7.99	6.81	7.99	8.65	$2.50 \cdot 10^{-6}$	$2.06 \cdot 10^{-6}$
amazon-2008	7.02	8.21	7.02	8.97	$1.36 \cdot 10^{-6}$	$2.05 \cdot 10^{-6}$
wiki-Vote	12.5	79.29	12.50	68.91	$1.21 \cdot 10^{-4}$	$6.83 \cdot 10^{-4}$
soc-Epinions1	6.71	34.29	6.71	32.47	$1.32 \cdot 10^{-5}$	$5.83 \cdot 10^{-5}$
soc-Slashdot0811	11.7	38.73	11.70	39.90	$1.21 \cdot 10^{-5}$	$3.06 \cdot 10^{-5}$
soc-Slashdot0902	11.54	39.08	11.54	40.48	$1.16 \cdot 10^{-5}$	$2.92 \cdot 10^{-5}$
soc-LiveJournal1	14.23	35.89	14.23	35.47	$2.07 \cdot 10^{-7}$	$5.69 \cdot 10^{-7}$
ljournal-2008	14.73	38.46	14.73	37.56	$1.88 \cdot 10^{-7}$	$5.10 \cdot 10^{-7}$
web-NotreDame	4.6	14.13	4.60	13.63	$3.11 \cdot 10^{-6}$	$1.25 \cdot 10^{-5}$
uk-2007-05@100K	30.51	139.95	30.51	46.40	$1.01 \cdot 10^{-5}$	$4.32 \cdot 10^{-5}$
cnr-2000	9.88	27.59	9.88	16.79	$3.12 \cdot 10^{-6}$	$9.08 \cdot 10^{-6}$
web-BerkStan	11.09	21.08	11.09	10.87	$1.46 \cdot 10^{-6}$	$3.36 \cdot 10^{-6}$
web-Google	5.57	17.80	5.57	9.24	$1.09 \cdot 10^{-6}$	$3.30 \cdot 10^{-6}$
in-2004	41.25	222.67	41.25	45.78	$1.01 \cdot 10^{-6}$	$5.06 \cdot 10^{-6}$
eu-2005	22.3	49.35	22.30	25.20	$1.16 \cdot 10^{-6}$	$2.99 \cdot 10^{-6}$
uk-2007-05@1M	12.23	41.71	12.23	19.61	$7.32 \cdot 10^{-7}$	$2.09 \cdot 10^{-6}$
indochina-2004	26.18	62.23	26.18	27.60	$1.37 \cdot 10^{-7}$	$4.09 \cdot 10^{-7}$
uk-2002	16.1	43.93	16.10	20.82	$5.48 \cdot 10^{-8}$	$1.43 \cdot 10^{-7}$
arabic-2005	28.14	82.68	28.14	34.96	$4.44 \cdot 10^{-8}$	$1.31 \cdot 10^{-7}$

Identifying “Connectivity Cuts”

Mislove et al [2007] tried to identify “vertex cuts” in social networks

Roughly, removing vertices of the “cut” breaks rest of the graph into many small, disconnected SCCs

Following approximation used in Web graph analysis, observed that after removing 10% of highest in (out) degree, largest SCC breaks into many smaller components

Can we exploit SAPs to do better? (still working on this)

Simpler question: how many of the high degree (removed) vertices are SAP?

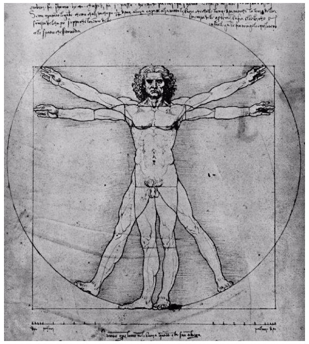
Identifying "Connectivity Cuts"

Graph	$\#sap_{scc}$	$\#sap_{scc}$ in $\delta^{-}(10\%)$	$\#sap_{scc}$ in $\delta^{+}(10\%)$
p2p-Gnutella04	1.3 K	132	69
enron	781	24	8
email-EuAll	960	3	5
wiki-Talk	14.8 K	53	152
amazon0302	69.6 K	2497	4801
amazon0312	69.0 K	998	3628
amazon-2008	102 K	374	1825
wiki-Vote	143	3	11
soc-Epinions1	8.1 K	89	90
soc-Slashdot0811	13.0 K	1	3
soc-Slashdot0902	14.1 K	2	5
soc-LiveJournal1	649 K	3057	3113
ljournal-2008	727 K	3729	3666
web-NotreDame	9.6 K	797	478
uk-2007-05@100K	2.8 K	24	153
cnr-2000	14.6 K	1587	741
web-BerkStan	53.6 K	1640	4179
web-Google	89.8 K	5150	4037
in-2004	33.5 K	2641	2907
eu-2005	99.3 K	10441	12928
uk-2007-05@1M	82.5 K	5825	8360
indochina-2004	503 K	30252	38724
uk-2002	1.8 M	152773	183997
arabic-2005	2.2 M	139907	176214

Outline of the Talk

1. Preliminary Definitions
2. Algorithms for strong articulation points and strong bridges
3. Preliminary Experiments on Large Scale Graphs
4. **Conclusions**

Summary



Presented linear-time algorithms to compute ***all*** strong articulation points and ***all*** strong bridges of directed graphs.

Theoretically optimal.

Intuitively, SAPs and SBs “connect” different groups / communities of directed networks

Summary

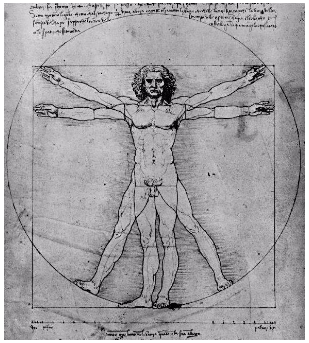
Fast in practice: first rough implementation on real-world large scale graphs able to process graphs with billions of edges in 10-15 minutes.

SAPs tend to appear frequently in real-world graphs (especially in co-product and social graphs). SBs tend to be less frequent

Both SAPs and SBs mostly concentrated in giant SCC

Avg in, out degree and PageRank of SAPs considerably higher than other vertices (further indication of their importance?)

Open Problems / Future Work

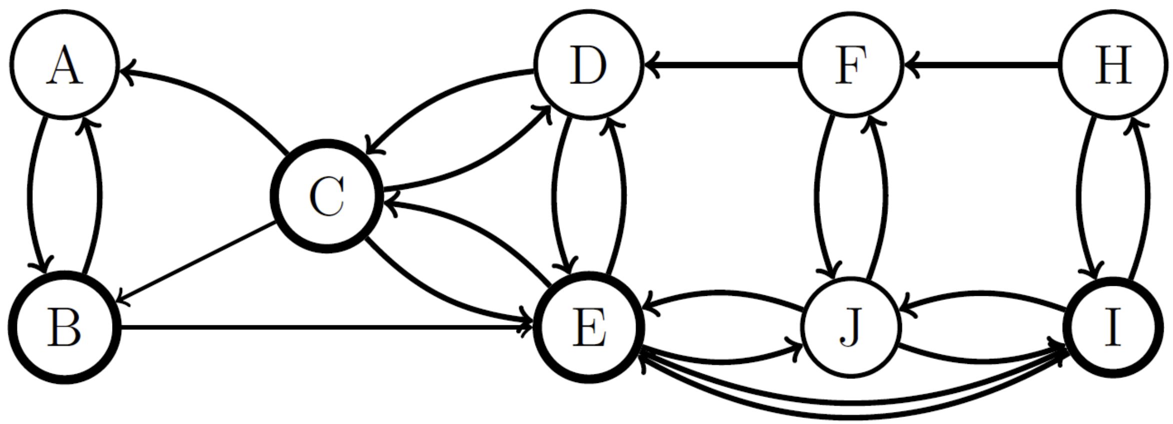


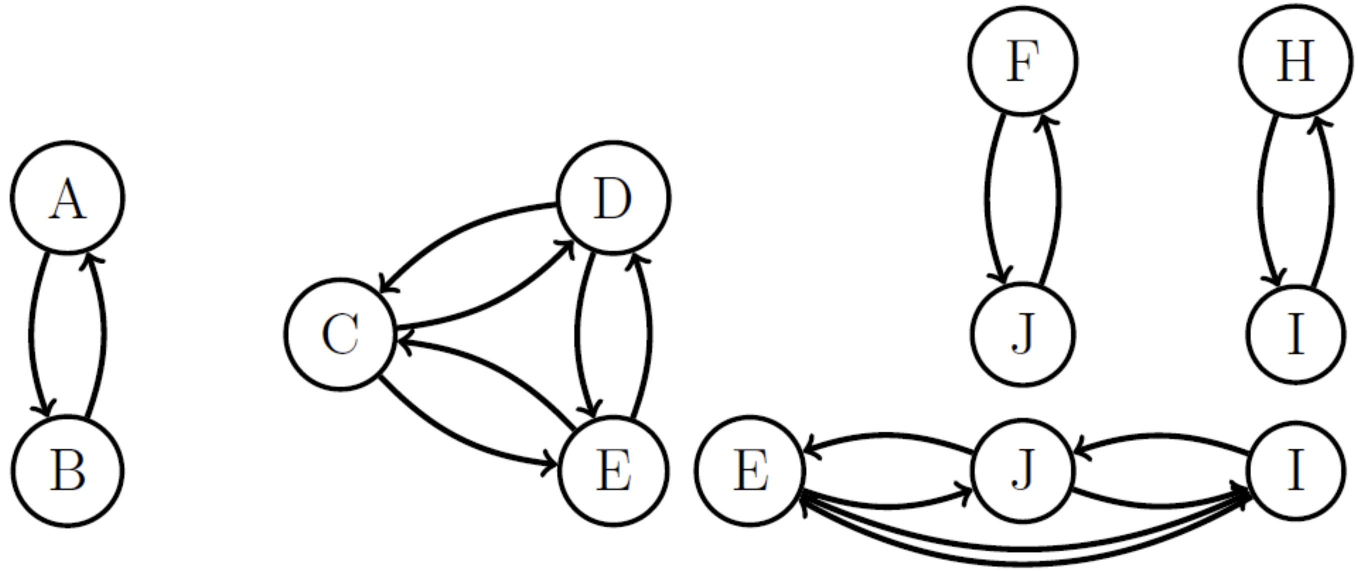
Higher connectivity cuts in strongly connected graphs? (e.g., separation pairs: vertex and edge cuts of cardinality 2)

Can the 2-vertex and 2-edge-connected components of a directed graph be computed in linear time?

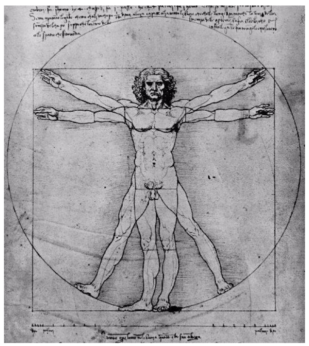
Best known time is $O(n(m+n))$ by repeatedly deleting SAPs / SBs.

2-vertex- and 2-edge-connected components are strange creatures





Open Problems / Future Work

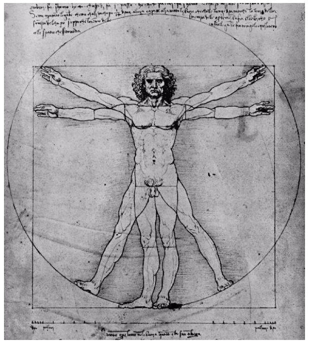


Higher connectivity cuts in strongly connected graphs? (e.g., separation pairs: vertex and edge cuts of cardinality 2)

Can the 2-vertex and 2-edge-connected components of a directed graph be computed in linear time?

> Best known time is *$O(n(m+n))$* by repeatedly deleting SAPs / SBs.

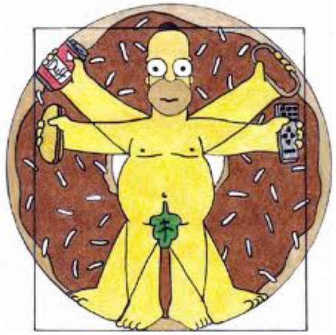
Open Problems / Future Work



Higher connectivity cuts in strongly connected graphs? (e.g., separation pairs: vertex and edge cuts of cardinality 2)

Can the 2-vertex and 2-edge-connected components of a directed graph be computed in linear time?

> Best known time is $O(n(m+n))$ by repeatedly deleting SAPs / SBs.



Perform more experiments to find "connectivity cuts" on social networks

Understand more the semantics of SAPs and SBs in real-world graphs