Script **generated by TTT**

Title: groh: profile1 (13.05.2015)

Date: Wed May 13 08:17:11 CEST 2015

Duration: 84:04 min

Pages: 76

Social Gaming / Social Computing SS 2015

PD Dr. Georg Groh

Applied Informatics / Cooperative Systems AICOS

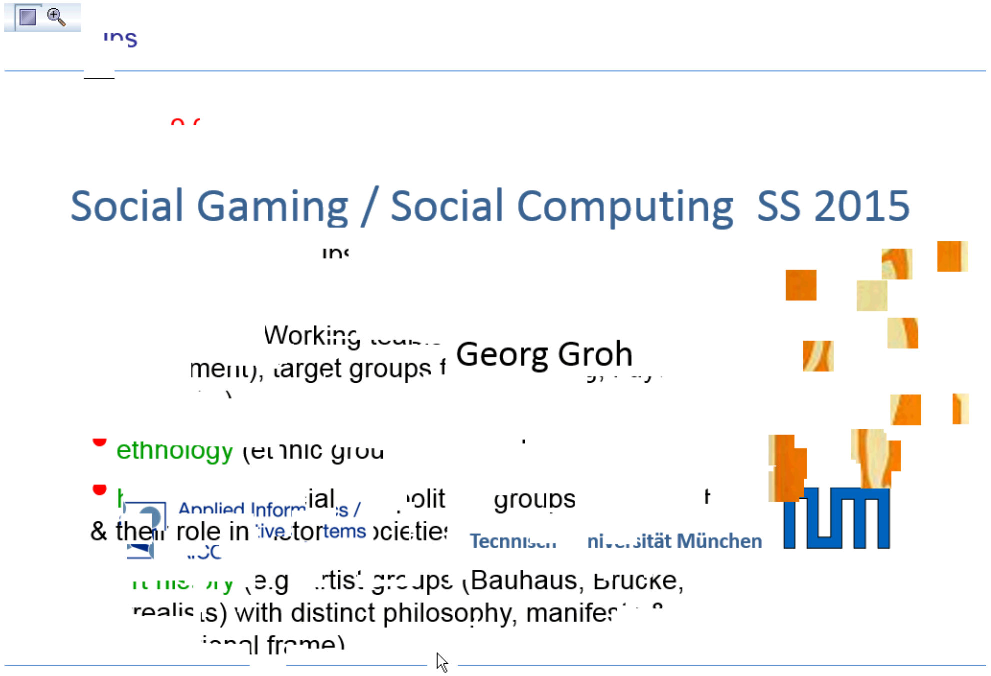
Technische Universität München

Social Gaming / Social Computing SS 2015

Working ... arget groups ... Georg Groh

Applied Informatics / Cooperative Systems AICOS

Technische Universität München



Social Gaming / Social Computing SS 2015

Working ... ment), target groups ... Georg Groh

- ethnology (ethnic grou...
- ...ial ... polit... groups ... & their role in ... societies

...e.g. artist groups (Bauhaus, Brücke, realis...s) with distinct philosophy, manifes... ...ional frame)

Applied Informatics / Cooperative Systems AICOS

Technische Universität München

ins

Social Computing SS 2015

Tönnis (18.. ,[3]:

1930s: Small group research (see [4,5,6])

Working ...

ment), target groups f...

Georg Groh

llec

- ethnology (ethnic grou...
- ...ial ...polit groups & their role in ... societies

Applied Informatics / Cooperative Systems

Technische Universität München

[6

...ry (e.g. artist groups (Bauhaus, Brücke, realists) with distinct philosophy, manifes... ...ional frame)

s, not a

---

ins

Social Computing SS 2015

Tönnis (18.. ,[3]:

1930s: Small group research (see [4,5,6])

istor

individual sch

tur in

Working ...

ment), ... dyad ...ot groups f...

Georg Groh

llec

- ethnology (ethnic group
- ...ial ... groups & their r... in ...

Technische Universität München

Homans (1950) [6]: "A group is a number ... communicate with one another often over ... that each person is able ... others, not at ... face."

(Bauhaus, Brücke, real...) with distinct philosophy, manifes... ...ional frame

---

ins

Social Computing SS 2015

Tönnis (18.. ,[3]:

1930s: Small group research (see [4,5,6])

istor

individual sch

tur in

Working ...

ment), ... dyad ...ot groups f...

Georg Groh

llec tio

- ethnology (ethnic group
- ...n view . Em... groups & their r... in ...

Technische Universität München

Homans (1950) [6]: "A group is a number ... communicate with one another often over ... span ... that each person is able ... others, not at ... face."

(Bauhaus, Brücke, real...) with di... philosophy, manifes...

---

ins

Social Computing SS 2015

Tönnis (18.. ,[3]:

1930s: Small group research (see [4,5,6]) ... are net...

istor

, comm...

individual sch

tur in

Working ...

ment), ... dyad ...ot groups f...

Georg Groh

llec tio

- ethnology (ethnic group
- ...n view . Em... groups & their r... in ...

Technische Universität München

Homans (1950) [6]: "A group is a number ... communicate with one another often over ... span ... that each person is able ... others, not at ... face."

(Bauhaus, Brücke, real...) with di... philosophy, manifes...

Georg Groh

- Tönnis (18.. [3]:
- 1930s: Small ... are netsearch (see [4,5,6])
- ...istic: com... , comm...
- individual ... al ro... nd ... group a... es
- ... ment), ... first groups ... (e.g. political party)
- ... group (e.g. family) ... seconda... group
- ethnology (ethnic group)
- ... n view ... groups
- & their ...
- Homans (1950) [6]: "A group is a number ... communicate with ... others ... (Bauhaus, Brucke, ... real ...) with ... that each person ... able to ... others, not at ... face."

Social Computing SS 2015

Georg Groh

Technische Universität München

- Tönnis (18.. [3]:
- 1930s: Small ... are netsearch (see [4,5,6])
- ...istic: com... , comm...
- individual ... al ro... nd ... group a... es
- ... ment), e.g. friends, groups ... groups (e.g. political party)
- primary group (e.g. family) ... seconda... group
- eth...
- ("my group") (special in group s r... hers") ... groups
- & their ...
- Homans (1950) [6]: "A group is a number ... communicate with ... others ... (Bauhaus, Brucke, ... real ...) that each person ... able to ... not at ... face."

Social Computing SS 2015

Georg Groh

Technische Universität München

- Tönnis (18.. [3]:
- 1930s: Small ... are netsearch ... [4,5,6] ... a
- ...istic: com... s, comm... , specia...
- individual ... al ro... and a... structure, group a... are es
- ... ment), e.g. friends, groups, cliques ←→ groups (e.g. political party)
- primary group (e.g. family) ... seconda... group
- eth...
- ("my group") (special in group s r... hers") ... groups
- & their ...
- Homans (1950) [6]: "A group is a number ... communicate with ... others ... (Bauhaus, Brucke, ... real ... masses", "that each person ... able to ... face ...."

Social Computing SS 2015

Georg Groh

Technische Universität München

- Tönnis (18.. [3]:
- 1930s: Small ... are netsearch ... [4,5,6] ... a
- ...istic: com... , spe... group
- individual ... al ro... members ... group a... are es
- ... ment), e.g. friends, groups, cliques ←→ groups (e.g. political party)
- primary group (e.g. family) ... seconda... group
- eth...
- ("my group") (special in ... up s r... hers") ... groups
- & their ...
- Homans (1950) [6]: "A group is a number ... communicate with ... others ... (Bauhaus, Brucke, ... real ... masses", "that each person ... able to ... face ...."

Tönnis (18…[3]:

Su… …ing SS …15

…1930s: … netserk … [4, … a

…cti… g… gr… grou…

…istc… com… spe…

…ividual … al ro… ments … members … group a … are … es

…tur… fri… ers)

…ment), g. friends … ← Groh groups (e.g. political party)

…ary … group (e.g. family … via s … onda … group

- eth… network analysis)

(“my group” … s r

- … hers” … groups

& their …

Homo… groups (F[6]: “A group is … nbe…

…om… … with … both … Bauh… is, Brucke,

…owea… mass”, “… that … person … le

…all”, ….

Tec… Universität München

Tönnis (18…[3]:

Su… …ing SS …15

…1930s: … netserk … [4, … a

…cti… …hood in … intra …

…istc… members … group a … es.

…tur… ers)

…ment), g. friends … ← Groh groups (e.g. political party)

…ary … group (e.g. family … via s … onda … group

- eth… network analysis)

(“my group” … s r

- … hers” … groups

& their …

Homo… groups (F[6]: “A group is … nbe…

…om… … with … both … Bauh… is, Brucke,

…owea… mass”, “… that … person … le

…all”, ….

Tec… Universität München

Tönnis (18…[3]:

Su… …ing SS …15

…1930s: … netserk … [4, … a

…cti… …borhood in … intra …

…istc… members … group a … es.

…ompactness: mean average … lengths are small

…ment), g. … connectivity is high … groups (e.g. political party)

…ary … group (e.g. family … via s … onda … group

- eth… network analysis)

(“my group” … s r

- … hers” … groups

& their …

Homo… groups (F[6]: “A group is … nbe…

…om… … with … both … Bauh… is, Brucke,

…owea… mass”, “… that … person … le

…all”, ….

Tec… Universität München

Tönnis (18…[3]:

Su… …ing SS …15

…1930s: … netserk … [4, … a

…cti… …borhood in … intra …

…istc… members … group a … es.

- ompactness: mean average … lengths are small

…os … and/or connectivity is high … groups (e.g. political party)

…intra … cluster coherence”

- eth… network analysis)

(“my group” … s r

- … he… groups

& their … group the…

Homo… groups (F[6]: “A group is … nbe…

…om… … with … both … Bauh… is, Brucke,

…owea… mass”, “… that … person … le

…all”, ….

Tec… Universität München

Social Computing SS 2015

Tönnis (18…[3]:

1930s:

neighborhood in G

intra cluster

compactness: mean average path lengths are small

groups and/or connectivity is high (e.g. political party)

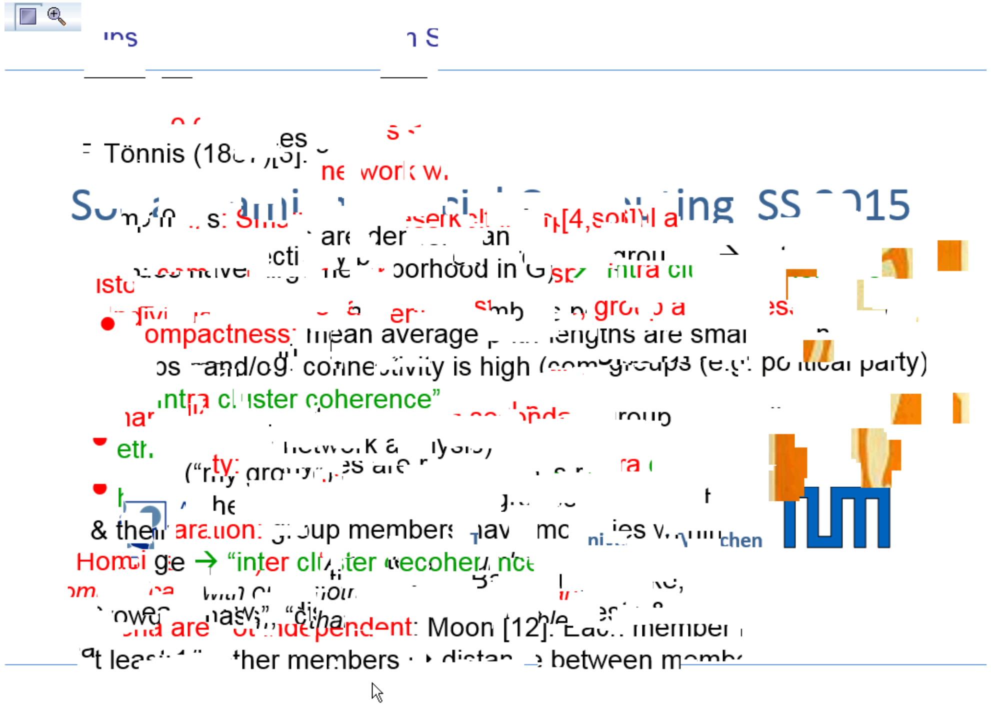
“intra cluster coherence”

network analysis

(“my group”)

separation: group members have more … ies with

Homophily → “inter cluster decoherence”



Social Computing SS 2015

Tönnis (18…[3]:

neighborhood in G

intra cluster

compactness: mean average path lengths are small

groups and/or connectivity is high (e.g. political party)

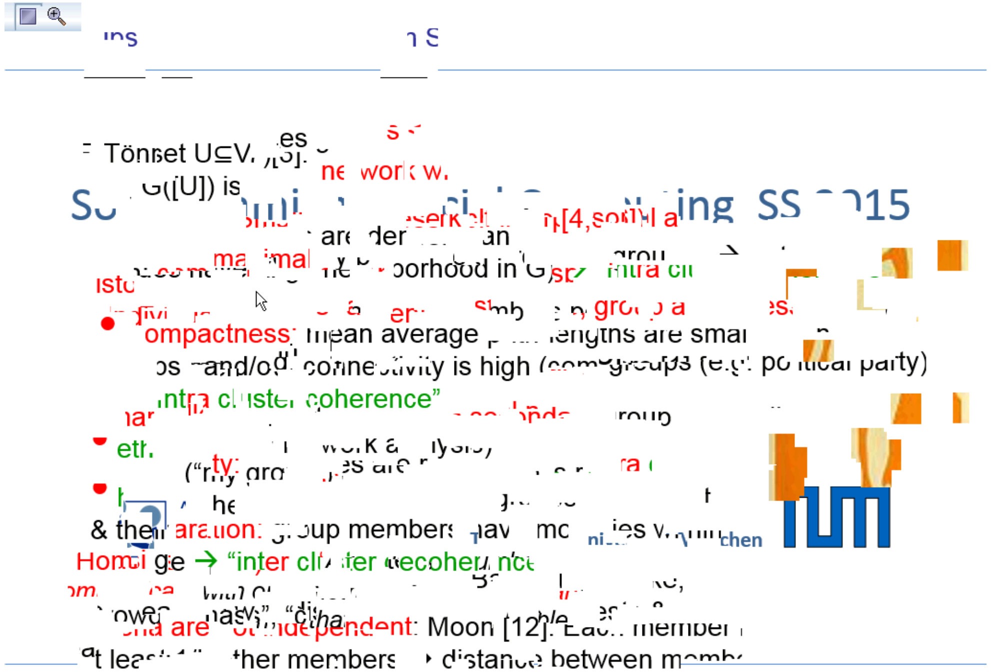
“intra cluster coherence”

separation: group members have more … ies with

Homophily → “inter cluster decoherence”

independent: Moon [12]: Each member

at least … other members … distance between members



Social Computing SS 2015

set U⊆V

G([U]) is

maximal

neighborhood in G

compactness: mean average path lengths are small

groups and/or connectivity is high (e.g. political party)

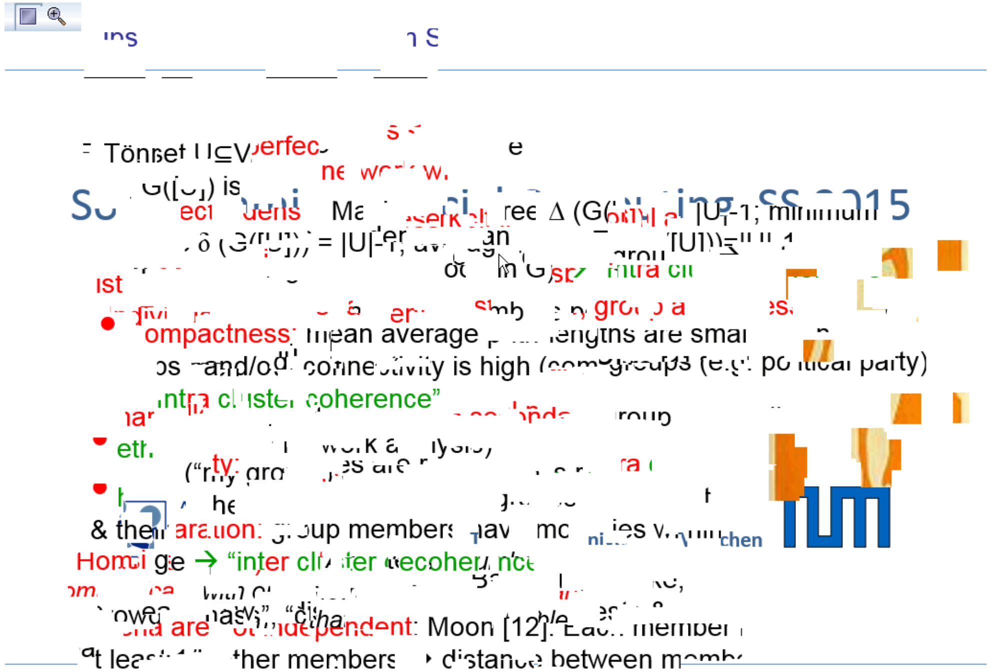
“intra cluster coherence”

separation: group members have more … ies with

Homophily → “inter cluster decoherence”

independent: Moon [12]: Each member

at least … other members … distance between members



Social Computing SS 2015

set U⊆V perfect

G([U]) is

Max … ree Δ (G([U])) = |U|-1; minimum

δ (G([U])) = |U|-1

compactness: mean average path lengths are small

groups and/or connectivity is high (e.g. political party)

“intra cluster coherence”

separation: group members have more … ies with

Homophily → “inter cluster decoherence”

independent: Moon [12]: Each member

at least … other members … distance between members

Tonset U⊆V perfec... e

G([U]) is ... network w...

... ree Δ (G([U])) = |U|-1; minimum

δ (G([U])) = |U|-1; average degree δ (G([U]))= ...

ist

- ...exactly co... members ... gn ... h le...s.
- compactness: mean average ... lengths are smal
- ...and/... connectivity is high ... subgroups ... e.g.: political party)
- intra cluster coherence”

etr.

(“rally group ... ty ...

& their ... aration: group members have ... then

Homog... ge → “inter cluster decoherence”

...ower ... easy”, ... independent: Moon [12]. Each member

at least ... other members ... distance between memb...

If |U|=k then G([U]) is ... (k-1) vertex- and edge-connected

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

G is n-edge connected if |V| ≥ 2 and G – Y is c... Y ⊆ E with |Y| < n;

SS 2015

$|E| > |V|^2/2\ (k-2)/(k-1)$ then

(k-1)

political party)

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

G is n-edge connected if |V| ≥ 2 and G – Y is connected for Y ⊆ E with |Y| < n;

SS 2015

$|V|^2/2\ (k-2)/(k-1)$ then

(k-1)

Each clique U

political party)

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

G is n-edge connected if |V| ≥ 2 and G – Y is connected for Y ⊆ E with |Y| < n;

SS 2015

$|V|^2/2\ (k-2)/(k-1)$ then

(k-1)

Each clique U

political party)

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

G is n-edge connected if |V| ≥ 2 and G – Y is connected for Y ⊆ E with |Y| < n;

SS 2015

$|V|^2/2\ (k-2)/(k-1)$ then

is N-clique

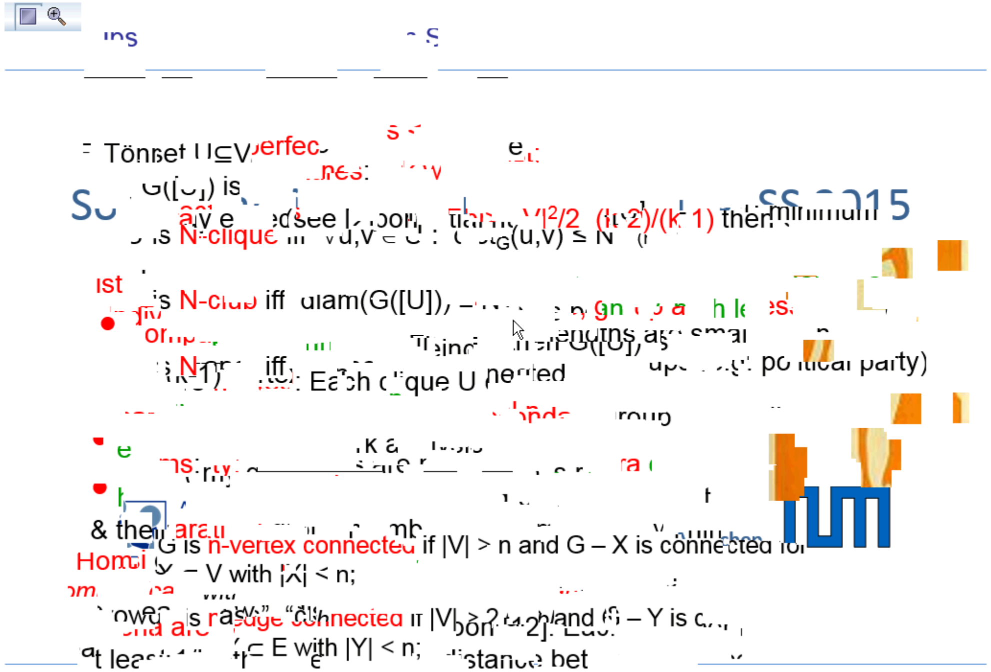
is N-clan iff

Each clique U

political party)

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

G is n-edge connected if |V| ≥ 2 and G – Y is connected for Y ⊆ E with |Y| < n;



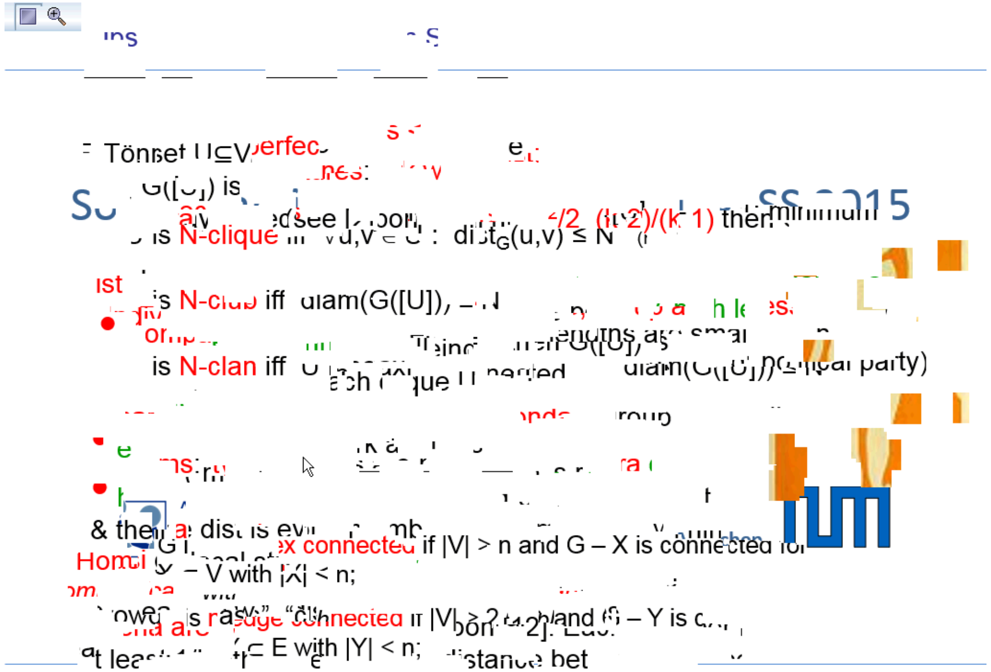
SS 2015

is N-clique iff ∀u,v ∈ U : $dist_G(u,v) \le N$

is N-club iff diam(G([U])) ≤ N

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

is n-edge connected if |V| ≥ 2 and G – Y is c… Y ⊆ E with |Y| < n;

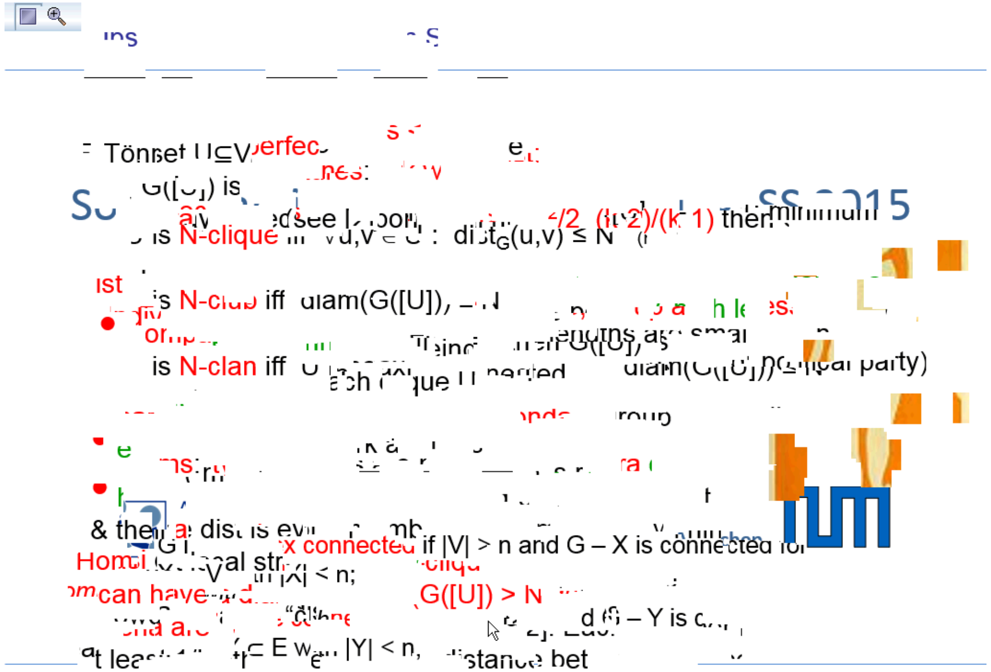


SS 2015

is N-clique iff ∀u,v ∈ U : $dist_G(u,v) \le N$

is N-clan iff U is N-clique and diam(G([U])) ≤ N

G is n-vertex connected if |V| > n and G – X is connected for X ⊆ V with |X| < n;

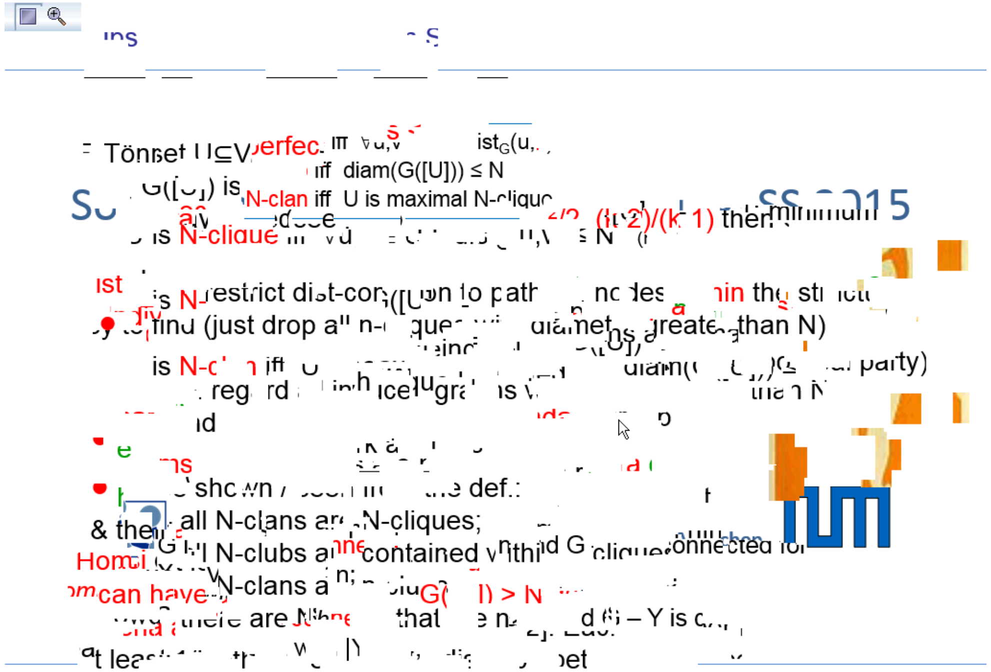


SS 2015

is N-clique iff ∀u,v ∈ U : $dist_G(u,v) \le N$

is N-clan iff U is N-clique and diam(G([U])) ≤ N

can have … G([U]) > N



SS 2015

iff diam(G([U])) ≤ N

N-clan iff U is maximal N-clique

restrict dist-comp to path nodes in the structure

find (just drop all N-cliques with diameter greater than N)

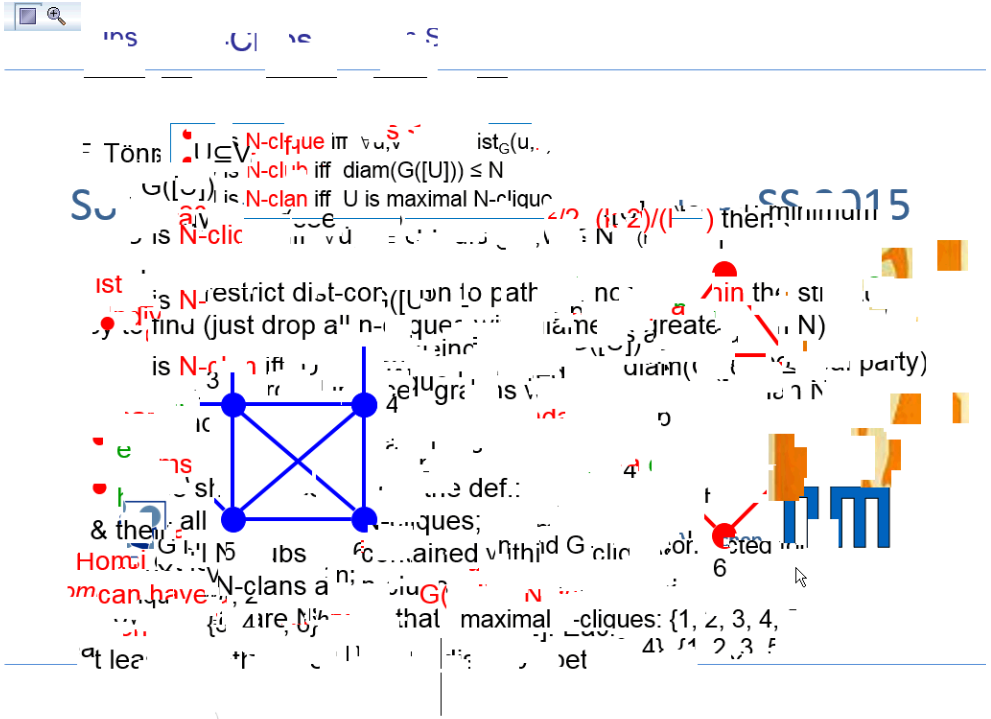
shown … the def.:

all N-clans are N-cliques;

all N-clubs are contained within N-clique

N-clans a…

can have … G([U]) > N



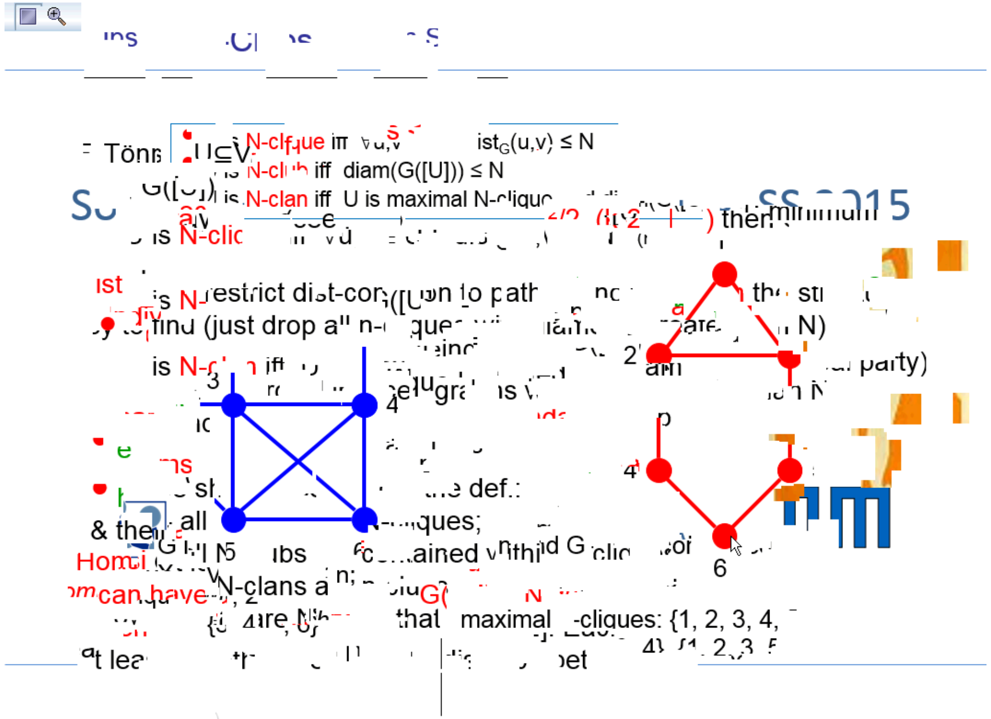
U⊆V is N-clique iff ∀u,v … $dist_G(u,$ …

U is N-club iff diam(G([U])) ≤ N

U is N-clan iff U is maximal N-clique

SS 2015

maximal 2-cliques: {1, 2, 3, 4,



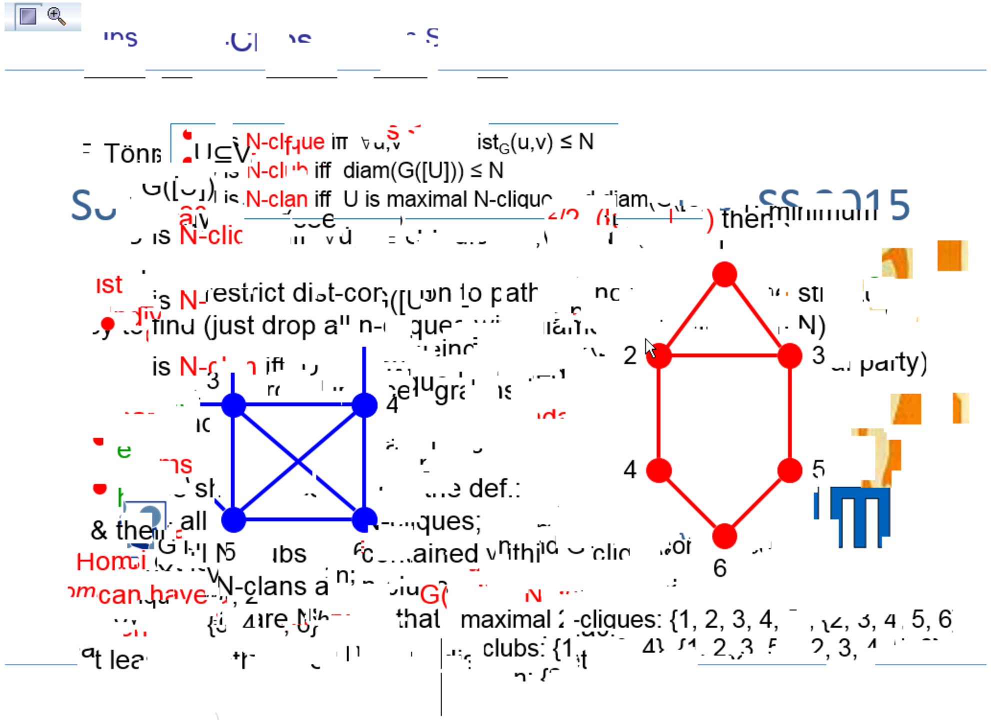
U⊆V is N-clique iff ∀u,v … $dist_G(u,v) \leq N$

U is N-club iff diam(G([U])) ≤ N

U is N-clan iff U is maximal N-clique

SS 2015

maximal 2-cliques: {1, 2, 3, 4,



U⊆V is N-clique iff ∀u,v … $dist_G(u,v) \leq N$

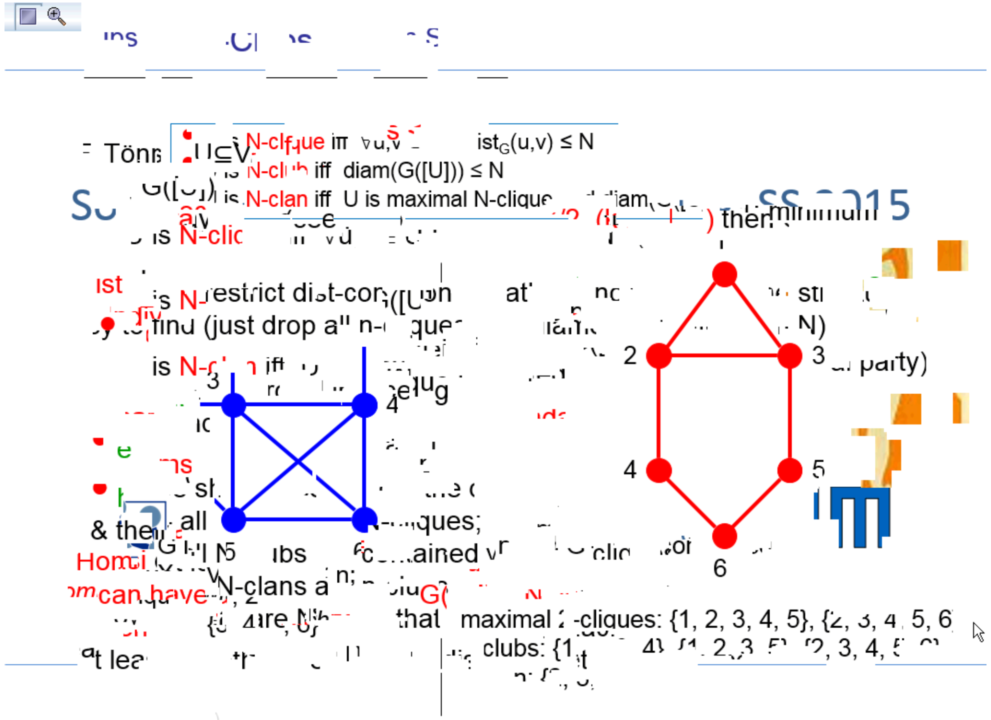
U is N-club iff diam(G([U])) ≤ N

U is N-clan iff U is maximal N-clique

SS 2015

maximal 2-cliques: {1, 2, 3, 4, … {2, 3, 4, 5, 6

2-clubs: {1, … {1, 2, 3, 5}, {2, 3, 4, …



U⊆V is N-clique iff ∀u,v … $dist_G(u,v) \leq N$

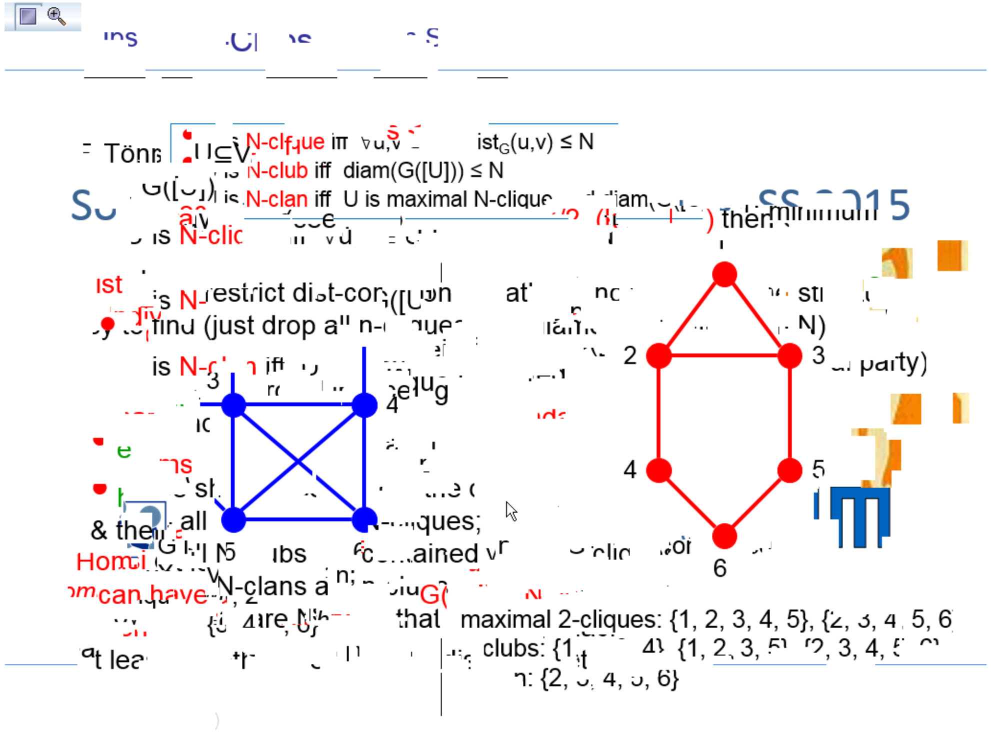
U is N-club iff diam(G([U])) ≤ N

U is N-clan iff U is maximal N-clique

SS 2015

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6

2-clubs: {1, … {1, 2, 3, 5}, {2, 3, 4, 5, 6}

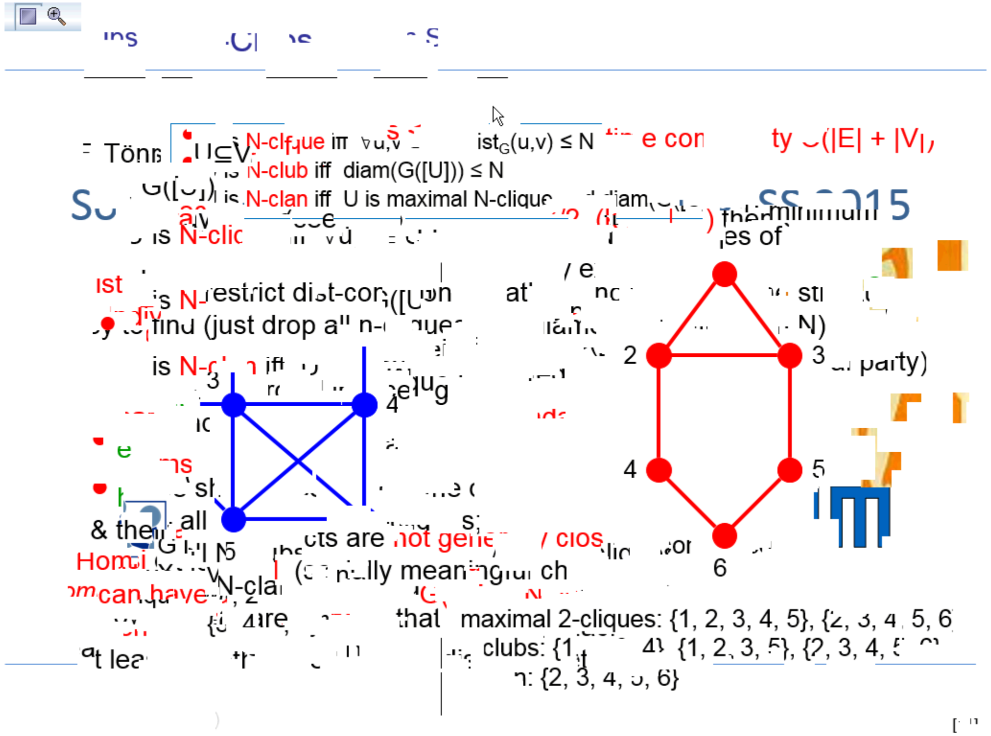


U is N-clique iff ∀u,v ∈ U: dist$_G$(u,v) ≤ N

U is N-club iff diam(G([U])) ≤ N

U is N-clan iff U is maximal N-clique

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}



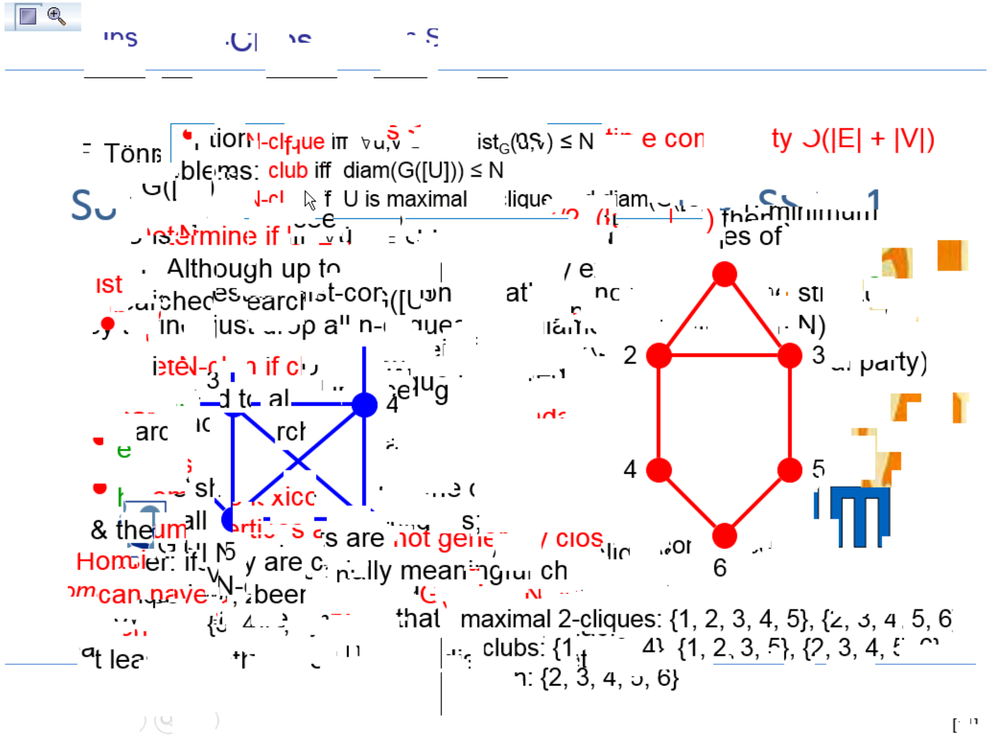
U is N-clique iff ∀u,v ∈ U: dist$_G$(u,v) ≤ N

U is N-club iff diam(G([U])) ≤ N

U is N-clan iff U is maximal N-clique

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

clubs: {1, 2, 3, 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}



U is N-clique iff ∀u,v ∈ U: dist$_G$(u,v) ≤ N

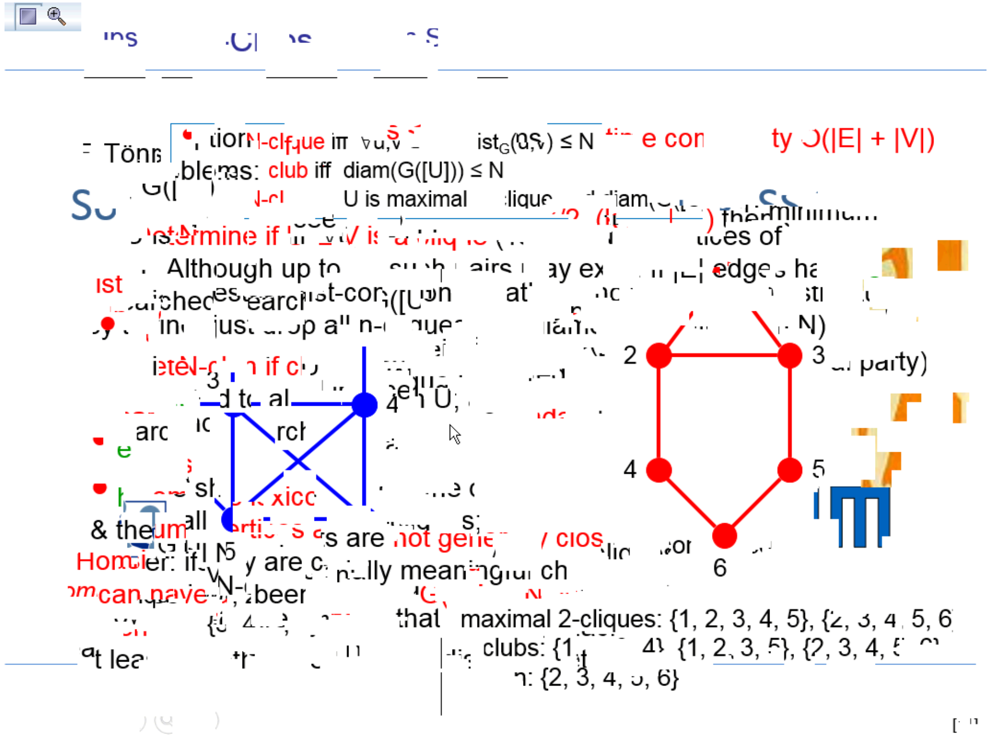
club iff diam(G([U])) ≤ N

U is maximal N-clique

Although up to

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

clubs: {1, 2, 3, 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}



U is N-clique iff ∀u,v ∈ U: dist$_G$(u,v) ≤ N

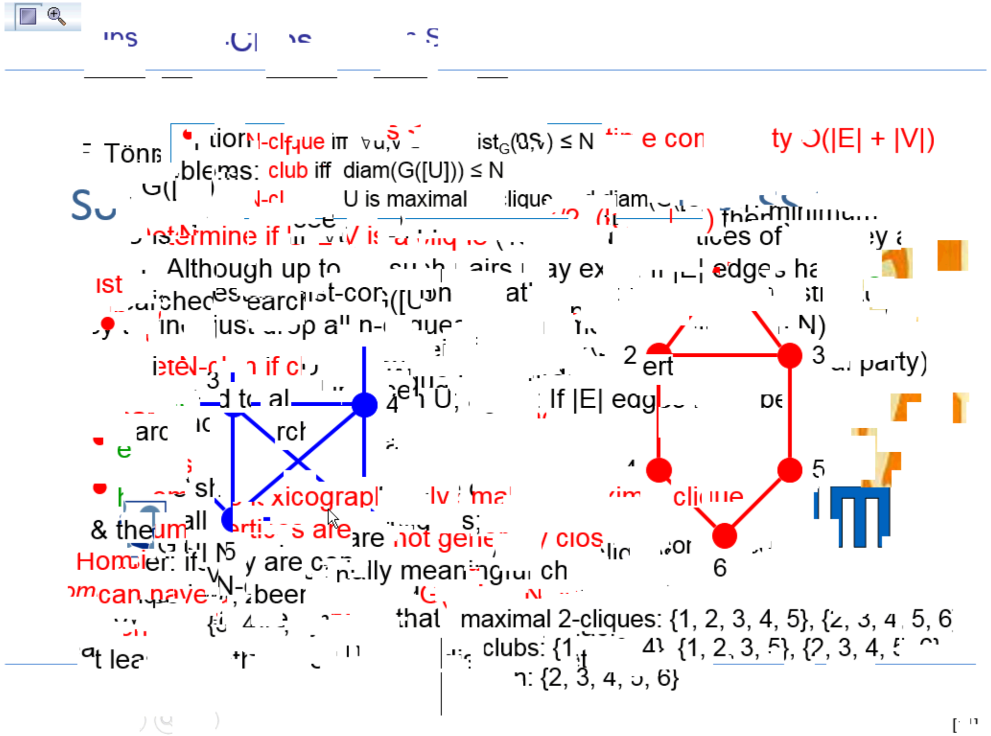
club iff diam(G([U])) ≤ N

U is maximal N-clique

Although up to

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

clubs: {1, 2, 3, 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}



N-clique iff ∀u,v ∈ U: dist_G(u,v) ≤ N … time complexity O(|E| + |V|)

club iff diam(G([U])) ≤ N

U is maximal … clique

Determine if … is a clique

Although up to … pairs … may exist … |E| edges …

If |E| equals …

(party)

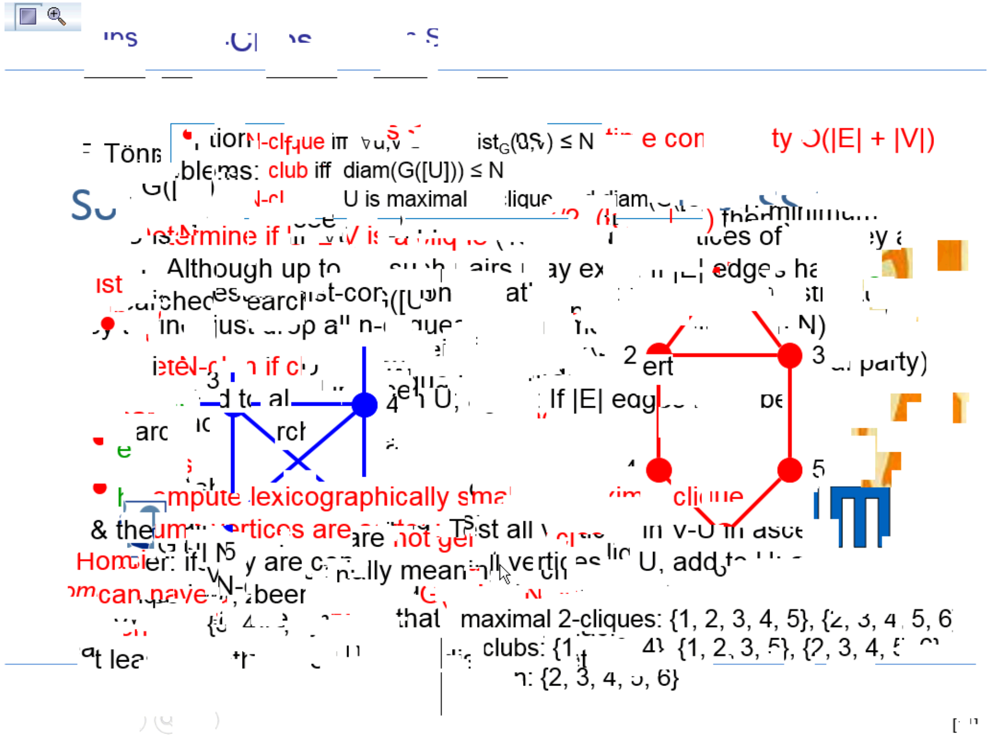
compute lexicographically smallest …

vertices are not generally closed …

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

clubs: {1, … 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}

n: {2, 3, 4, 5, 6}



N-clique iff ∀u,v ∈ U: dist_G(u,v) ≤ N … time complexity O(|E| + |V|)

club iff diam(G([U])) ≤ N

U is maximal … clique

Determine if … is a clique

Although up to … pairs … may exist … |E| edges …

If |E| equals …

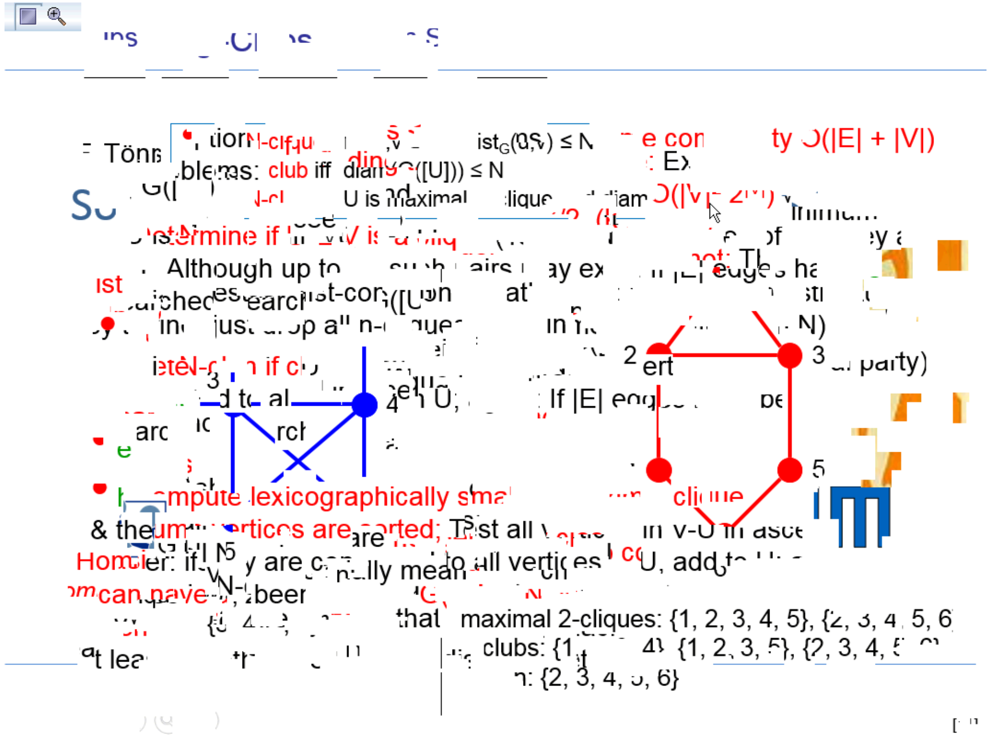
Compute lexicographically smallest …

& the … vertices are not generally …: Test all vertices … in V-U in ascending order … U, add to …

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

clubs: {1, … 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}

n: {2, 3, 4, 5, 6}



N-clique iff ∀u,v ∈ U: dist_G(u,v) ≤ N … time complexity O(|E| + |V|)

club iff diam(G([U])) ≤ N

Exhaustive search: O(|V|² 2^|V|)

U is maximal … clique

Determine if … is a clique

Although up to … pairs … may exist … |E| edges …

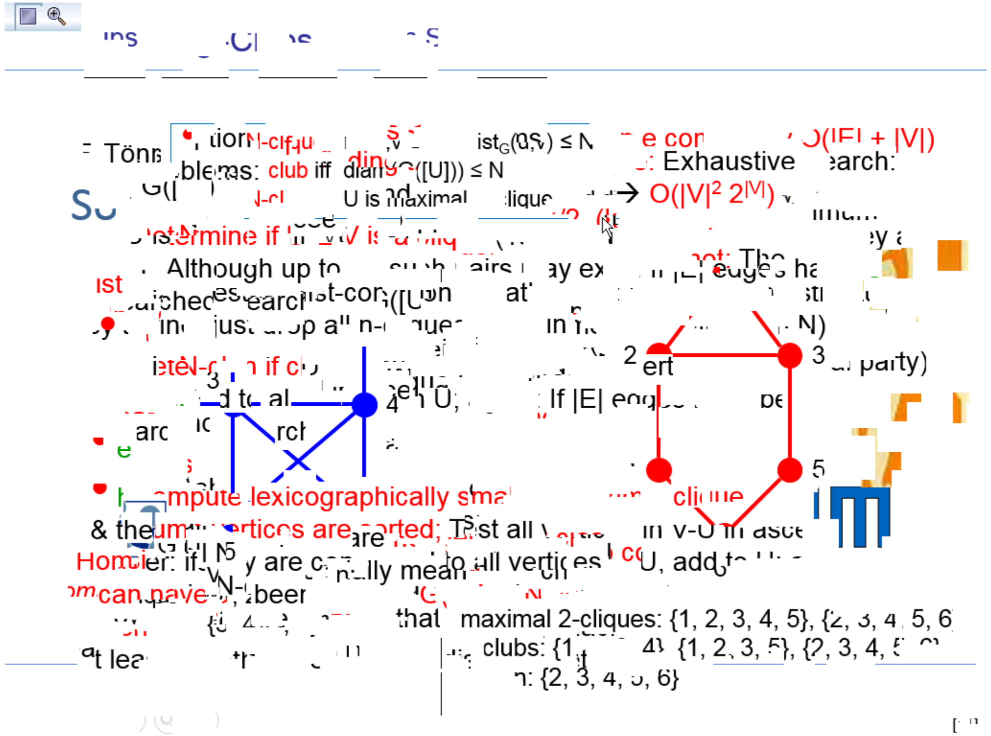
Compute lexicographically smallest …

& the … vertices are sorted; Test all vertices … in V-U in ascending order … to all vertices … U, add to …

maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

clubs: {1, … 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}

n: {2, 3, 4, 5, 6}



N-clique iff ∀u,v ∈ U: dist_G(u,v) ≤ N … time complexity O(|E| + |V|)

club iff diam(G([U])) ≤ N

Exhaustive search: → O(|V|² 2^|V|)

U is maximal … clique

Determine if … is a clique

Although up to … pairs … may exist … |E| edges …

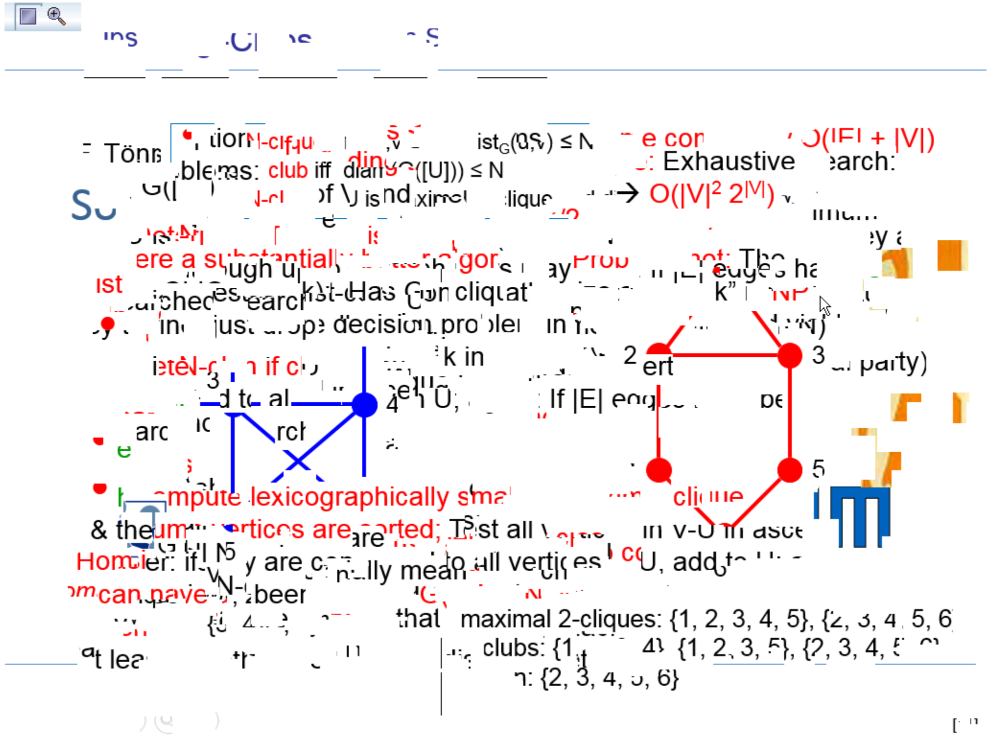
Compute lexicographically smallest …

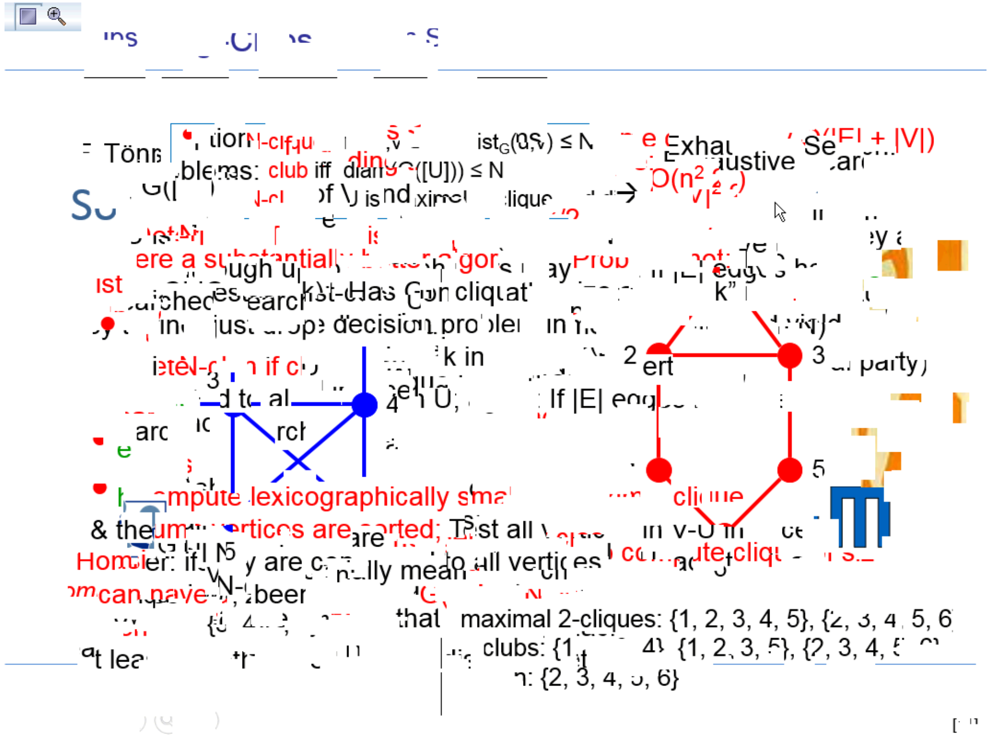
& the … vertices are sorted; Test all vertices … in V-U in ascending order … to all vertices … U, add to …

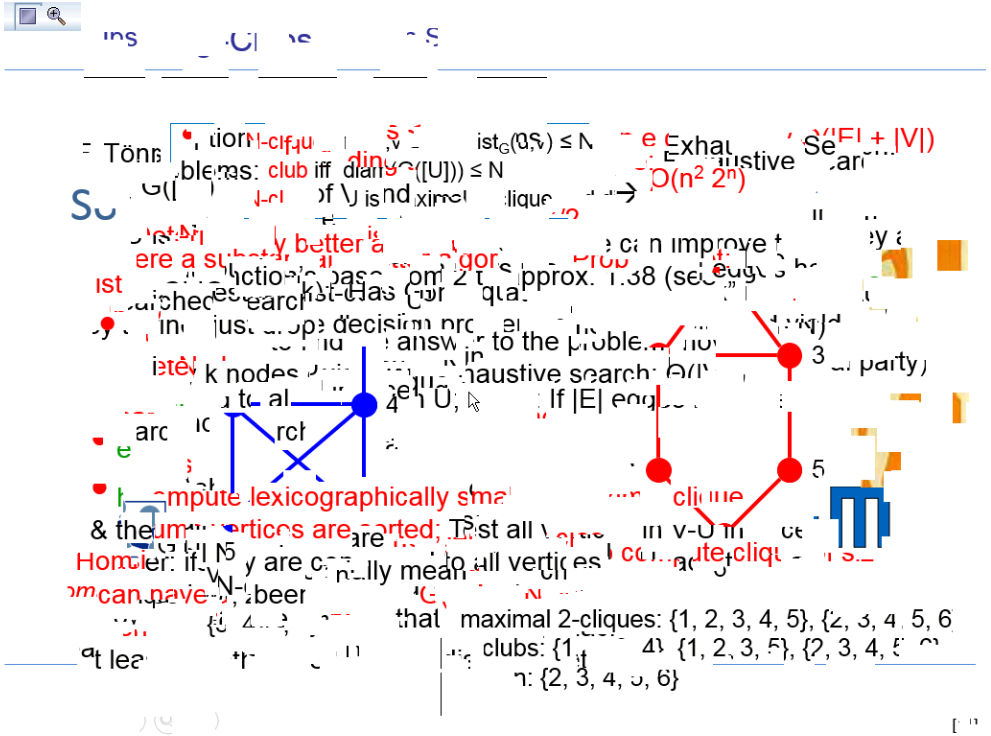
maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

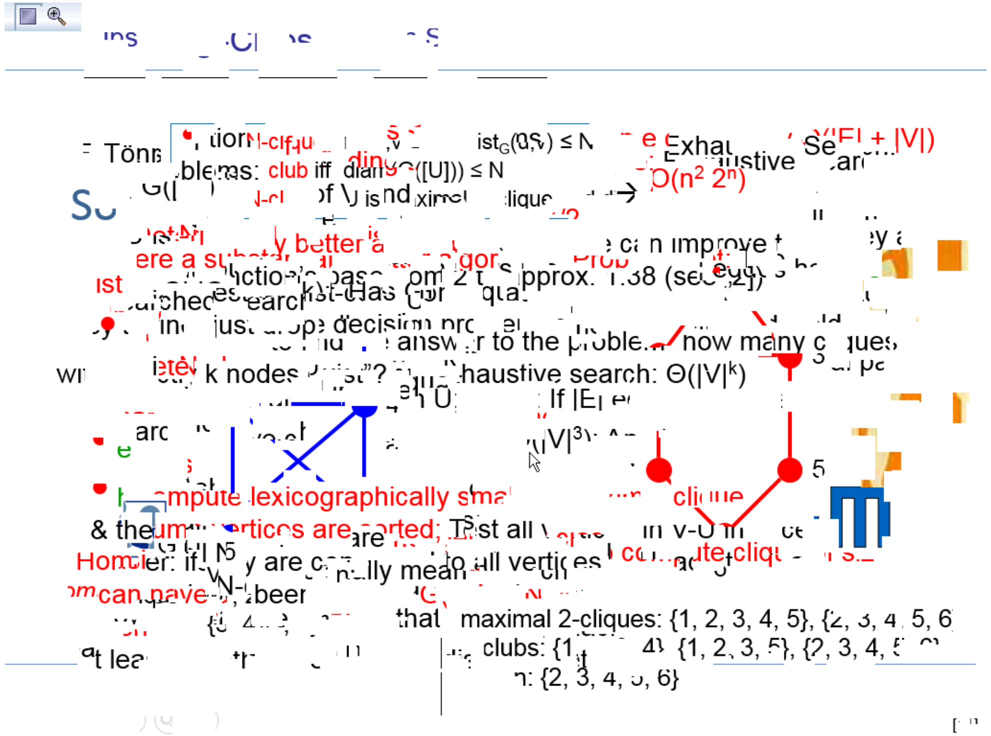
clubs: {1, … 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}

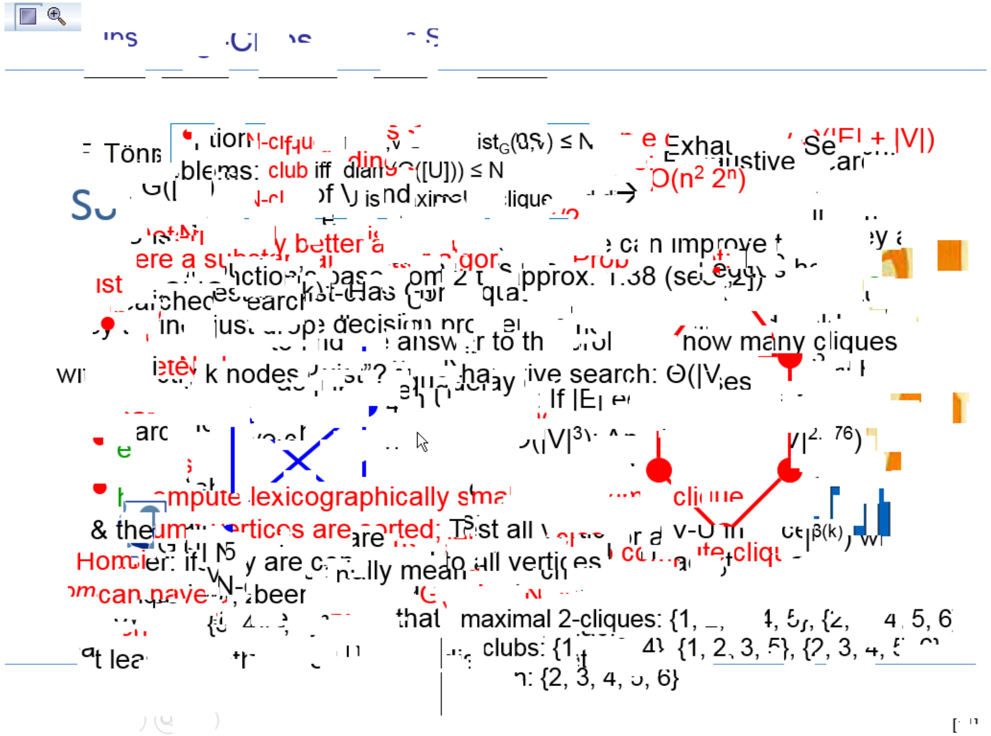
n: {2, 3, 4, 5, 6}

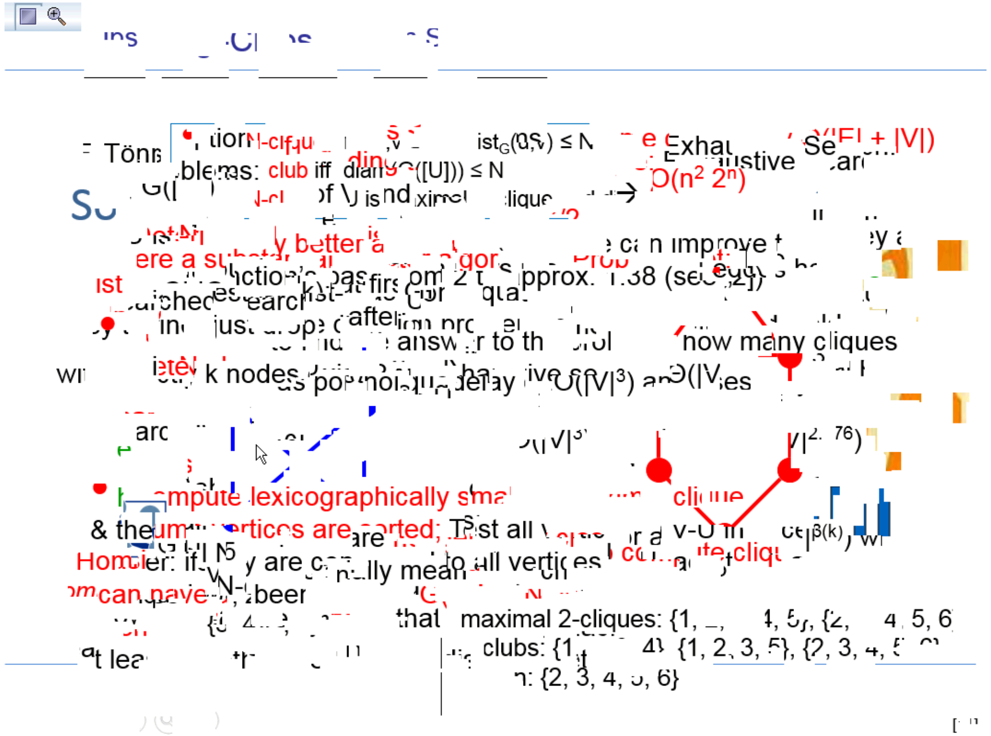

Exhaustive search:
O(|V|² 2^|V|)
If |E| equals
maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6
{2, 3, 4, 5, 6}

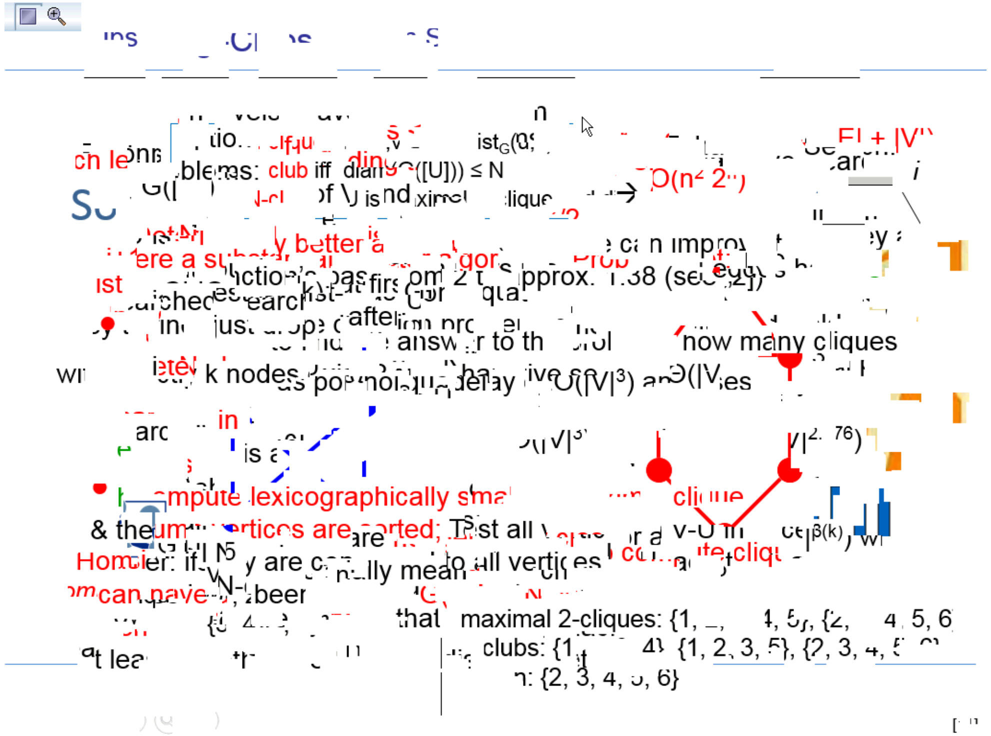

Exhaustive Search
maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6
{2, 3, 4, 5, 6}

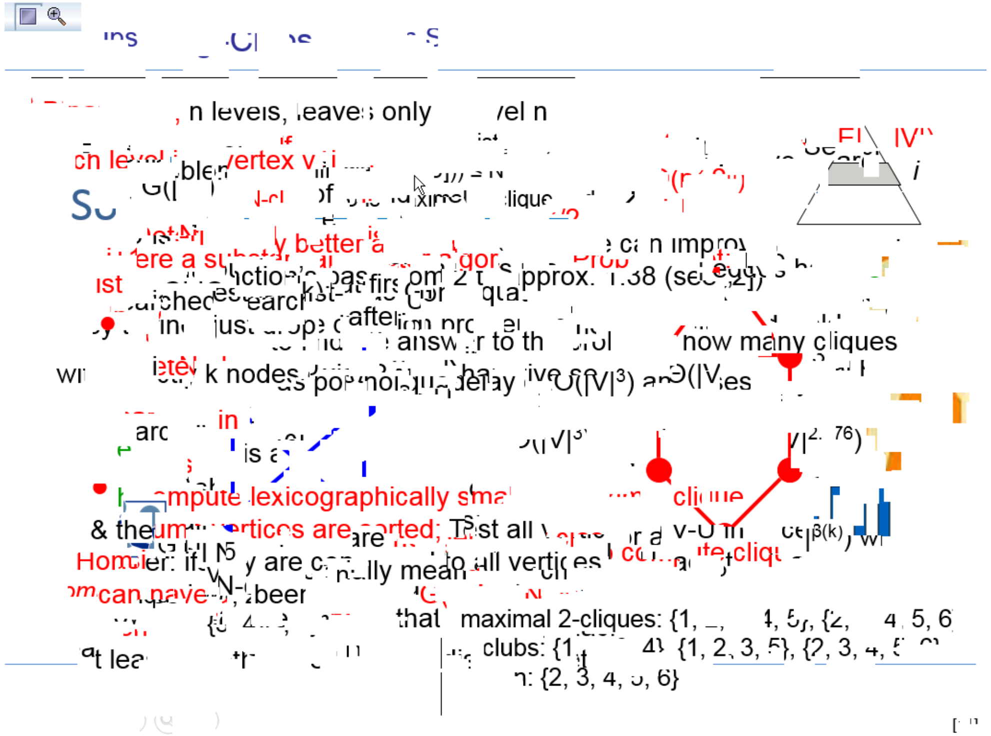

Exhaustive Search
O(n² 2ⁿ)
approx. 1.38
maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6
{2, 3, 4, 5, 6}

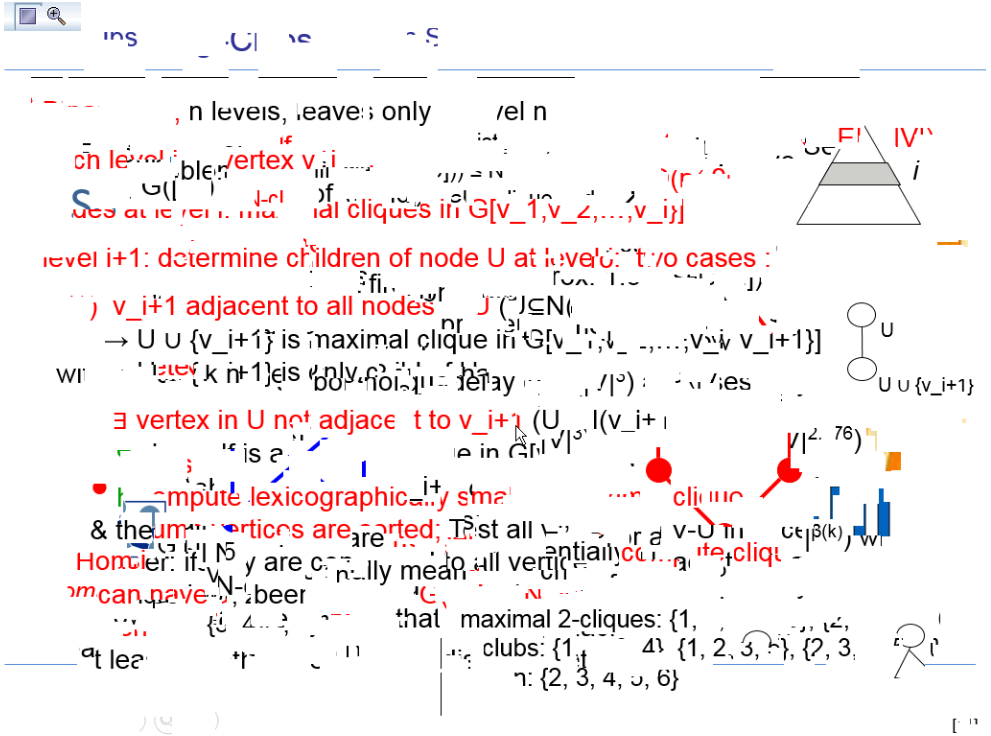

Exhaustive Search
O(n² 2ⁿ)
approx. 1.38
how many cliques
k nodes
haustive search: Θ(|V|^k)
maximal 2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6
{2, 3, 4, 5, 6}











, n levels, leaves only … vel n

ch level … vertex v_i

… at level i: maximal cliques in G[v_1,v_2,…,v_i]

level i+1: determine children of node U at level i: two cases :

) v_i+1 adjacent to all nodes … U (U⊆N(…

→ U ∪ {v_i+1} is maximal clique in G[v_1,v_2,…,v_i, v_i+1]

∃ vertex in U not adjacent to v_i+1 (U … (v_i+1 …

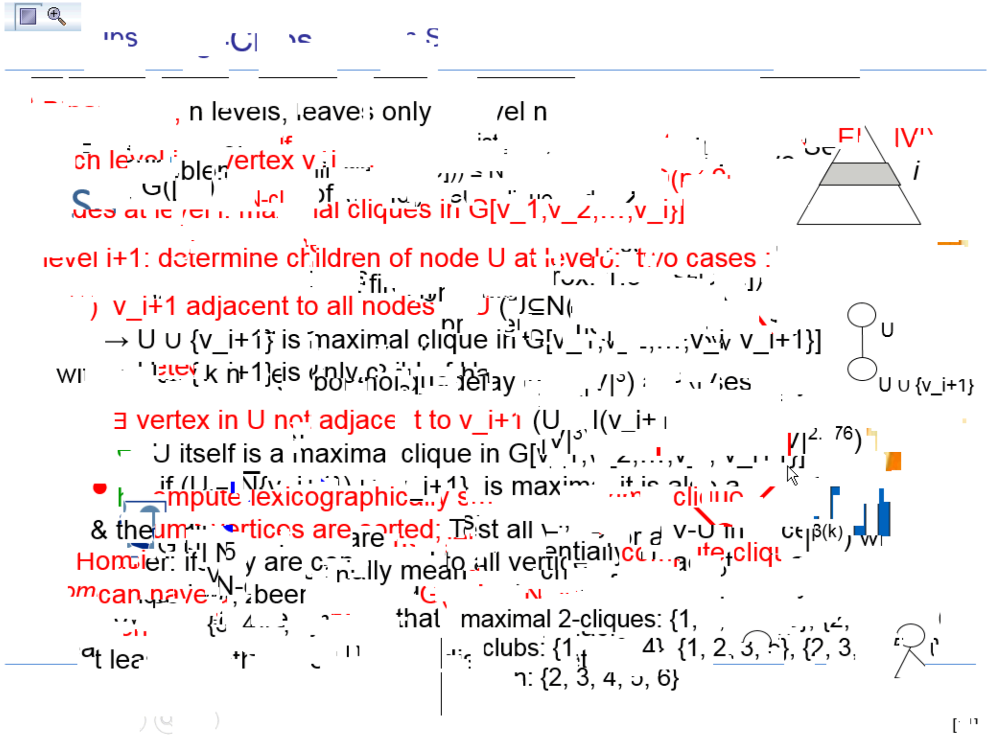
compute lexicographically sma… clique

& the … vertices are sorted; first all …

maximal 2-cliques: {1, …

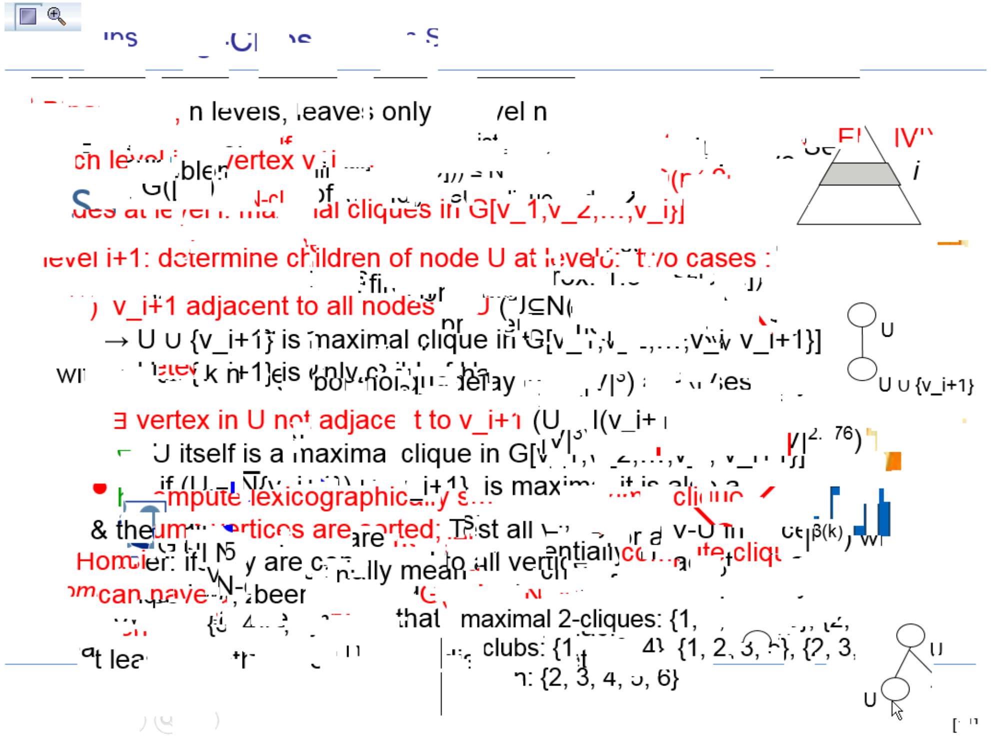
clubs: {1, … 4} {1, 2, 3, …}, {2, 3,

n: {2, 3, 4, 5, 6}

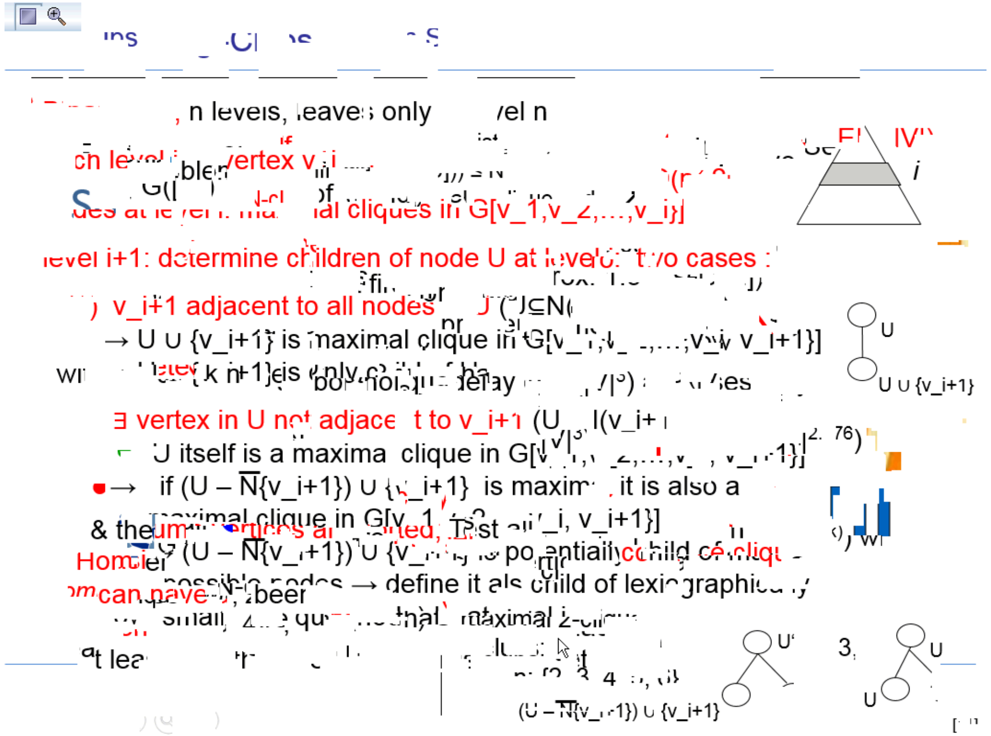


U itself is a maximal clique in G[v_1, v_2,…,v_i, v_i+1]

if (U – N̄{v_i+1}) ∪ {v_i+1} is maxim… it is also a

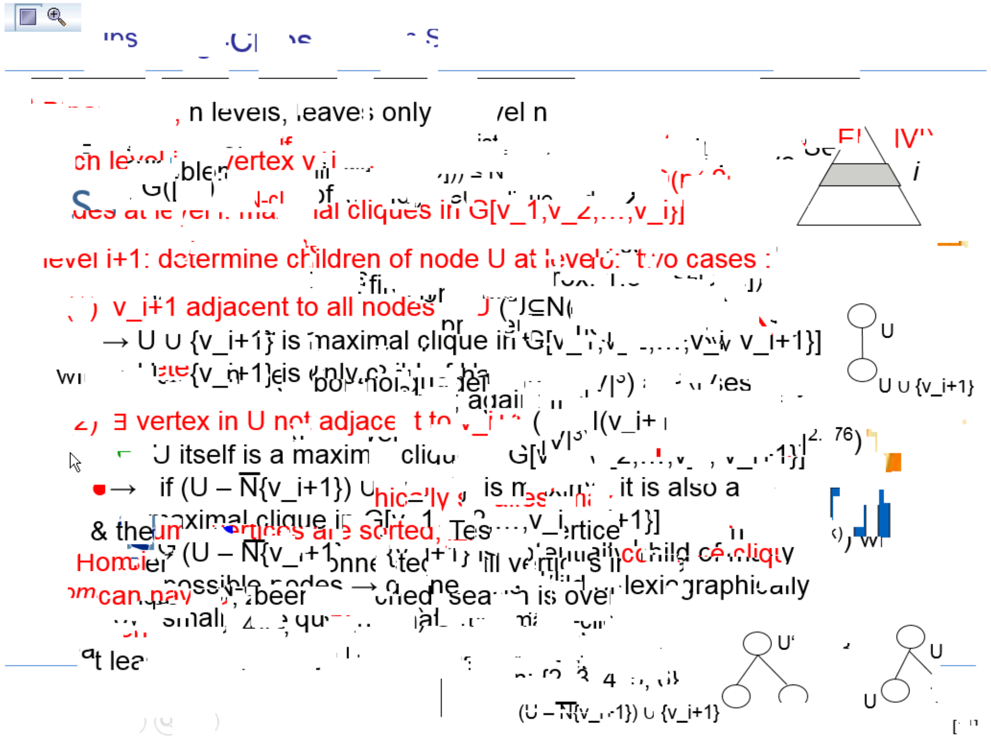


U itself is a maximal clique in G[v_1, v_2,…,v_i, v_i+1]



→ if (U – N̄{v_i+1}) ∪ {v_i+1} is maxim… it is also a maximal clique in G[v_1, v_2, …, v_i, v_i+1]

(U – N̄{v_i+1}) ∪ {v_i+1} is potentially child of … possible nodes → define it als child of lexicographically small…



, n levels, leaves only … vel n

ch level … vertex v_i

des at le… …al cliques in G[v_1,v_2,…,v_i]

level i+1: determine children of node U at level i: two cases :

(1) v_i+1 adjacent to all nodes … U ⊆ N(

→ U ∪ {v_i+1} is maximal clique in G[v_1,v_2,…,v_i, v_i+1]

… {v_i+1} is only c… …der

again

(2) ∃ vertex in U not adjacent to v_i+1 … I(v_i+1

→ U itself is a maximal clique … G[v_1,v_2,…,v_i, v_i+1]

→ if (U – N̄{v_i+1}) ∪ … is maximal … it is also a maximal clique in G[v_1,…,v_i+1]

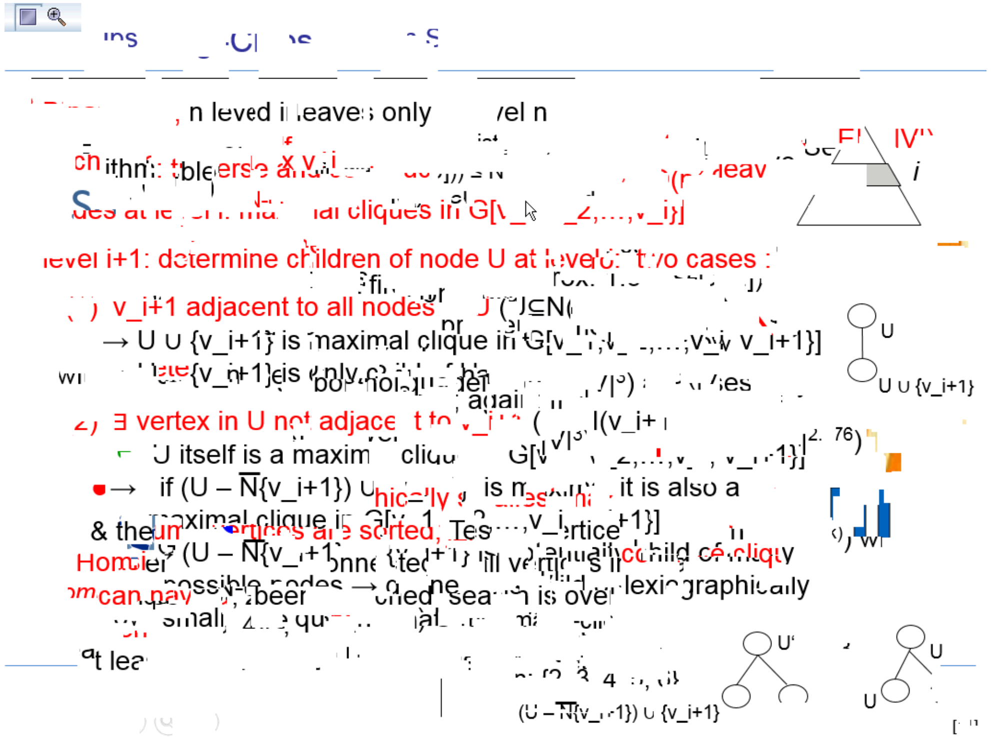
& the … sorted, Tes… …rtice

Homier … (U – N̄{v_i+1}) … child of cliqu

… possible nodes → … lexicographically

small …

at lea…



, n level i leaves only … vel n

ithm: traverse and … vertex v_i

des at le… …al cliques in G[v_1,…,v_i]

level i+1: determine children of node U at level i: two cases :

(1) v_i+1 adjacent to all nodes … U ⊆ N(

→ U ∪ {v_i+1} is maximal clique in G[v_1,v_2,…,v_i, v_i+1]

(2) ∃ vertex in U not adjacent to v_i+1 … I(v_i+1

→ U itself is a maximal clique … G[v_1,v_2,…,v_i, v_i+1]

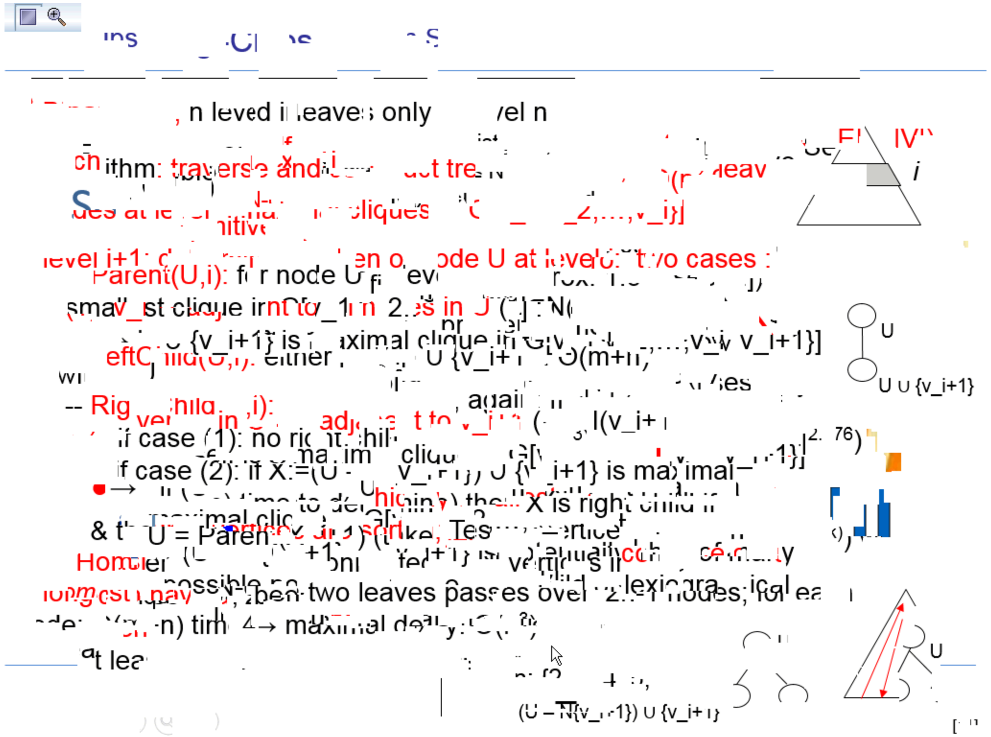
→ if (U – N̄{v_i+1}) ∪ … is maximal … it is also a maximal clique in G[v_1,…,v_i+1]

& the … sorted, Tes… …rtice

Homier … (U – N̄{v_i+1}) … child of cliqu

… possible nodes → … lexicographically

at lea…



, n level i leaves only … vel n

ithm: traverse and … tree … leav

des at le… …al cliques … G[v_1,…,v_i]

level i+1: … node U at level i: two cases :

Parent(U,i): for node U … level

smallest clique in … v_1 … 2 … in … N(

… ∪ {v_i+1} is maximal clique in G[v_1,…,v_i, v_i+1]

LeftChild(U,i): either … U ∪ {v_i+1} … O(m+n)

-- Right Child(U,i):

if case (1): no right child

if case (2): if X:=(U – N̄{v_i+1}) ∪ {v_i+1} is maximal

→ … X is right child …

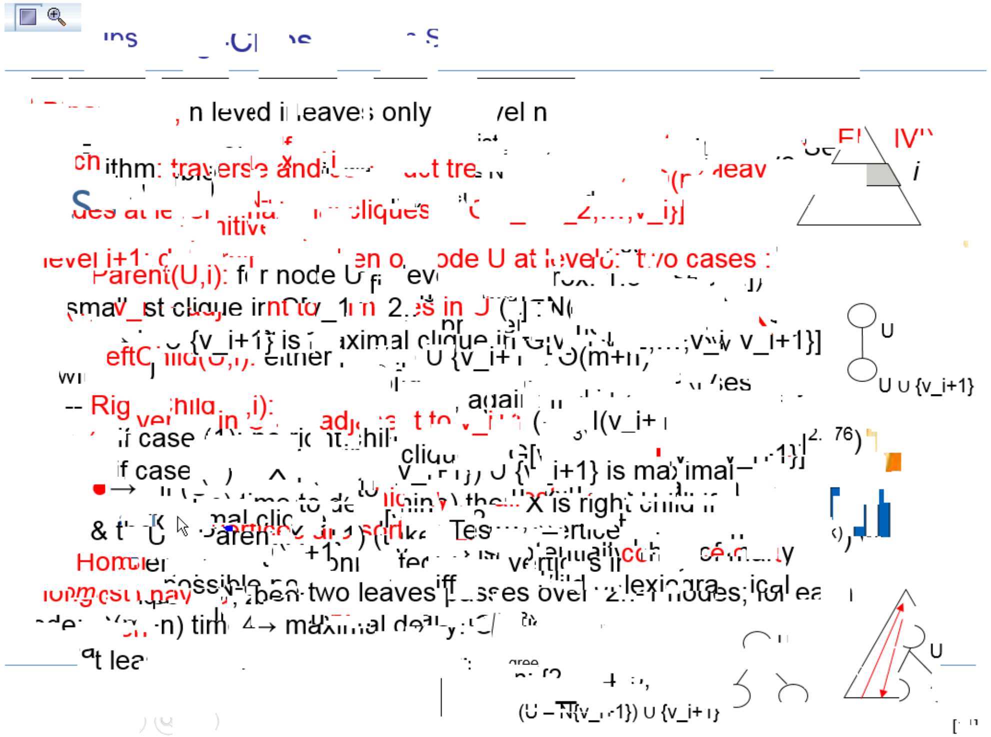
U = Parent(X, i+1) …

Homier …

… between two leaves passes over … 2n nodes, for each … lexicographical

… O(m+n) time → maximal delay: O(…)

at lea…



, n level i leaves only … vel n

ithm: traverse and … tree … leav

des at le… …al cliques … G[v_1,…,v_i]

level i+1: … node U at level i: two cases :

Parent(U,i): for node U … level

smallest clique in … v_1 … 2 … in … N(

LeftChild(U,i): either … U ∪ {v_i+1} … O(m+n)

-- Right Child(U,i):

if case … no right child

if case …

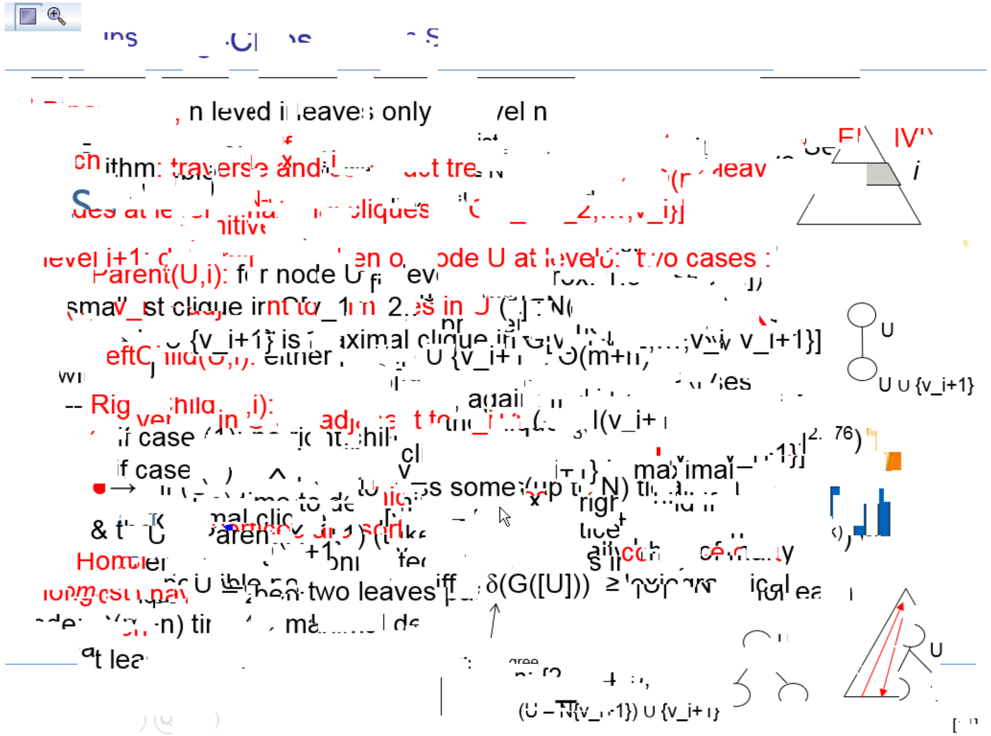
… X is right child …

Homier …

… between two leaves passes over … 2n nodes, for each … lexicographical

… O(m+n) time → maximal delay: O(…)

at lea…



n leved i leaves only vel n

ithm: traverse and ut tree leav

des at le itive cliques {v_1, v_2,...,v_i}]

level i+1: en o ode U at level i: two cases :

Parent(U,i): for node U every

smallest clique in 2. is in

{v_i+1} is aximal clique in G[v v_i+1}]

LeftChild(U,i): either U ∪ {v_i+1} O(m+n) ves

-- RightChild(U,i): adj t to N(v_i+1)

if case child

if case some maximal

→ right

& t parent (v_i+1)

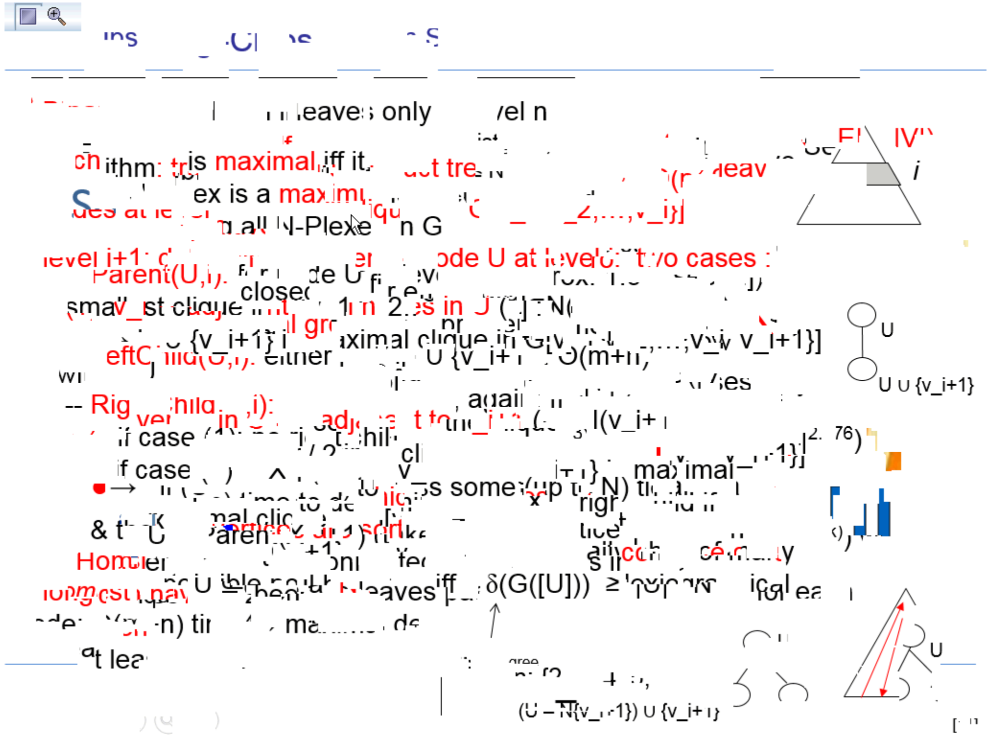
Homer

two leaves iff δ(G([U])) ≥

n) tim ma ds

at lea

(U – N(v_i+1)) ∪ {v_i+1}



leaves only vel n

ithm: is maximal iff it ut tree leav

ex is a maximal iqu {v_1, v_2,...,v_i}]

all N-Plexe n G

level i+1: ode U at level i: two cases :

Parent(U,i): closed de U every

smallest clique in 2. is in

al gr {v_i+1} aximal clique in G[v v_i+1}]

LeftChild(U,i): either U ∪ {v_i+1} O(m+n) ves

-- RightChild(U,i): adj t to N(v_i+1)

if case child

if case some maximal

→ right

& t parent (v_i+1)

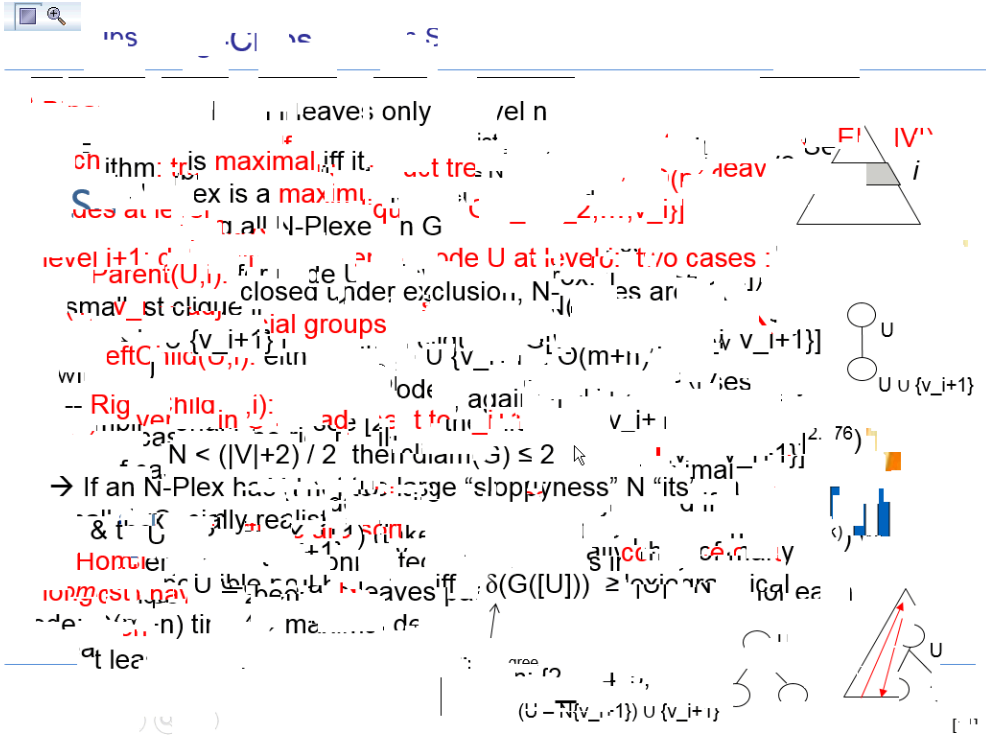
Homer

leaves iff δ(G([U])) ≥

n) tim ma ds

at lea

(U – N(v_i+1)) ∪ {v_i+1}



leaves only vel n

ithm: is maximal iff it ut tree leav

ex is a maximal iqu {v_1, v_2,...,v_i}]

all N-Plexe n G

level i+1: ode U at level i: two cases :

Parent(U,i): closed under exclusion, N- es are

smallest clique ial groups

LeftChild(U,i): eith U ∪ {v_ O(m+n) ves

-- RightChild(U,i): ode again

N < (|V|+2) / 2 then diam(G) ≤ 2

→ If an N-Plex has large “sloppyness” N “its”

& t ally realist

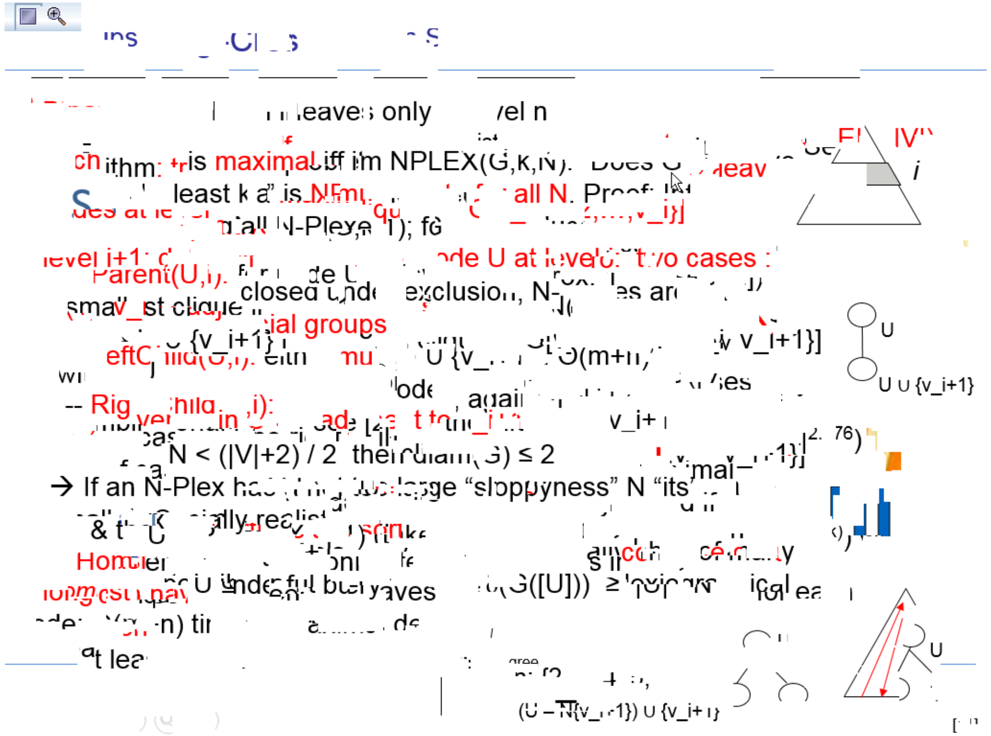
Homer

leaves iff δ(G([U])) ≥

n) tim ma ds

at lea

(U – N(v_i+1)) ∪ {v_i+1}



leaves only vel n

ithm: is maximal iff im NPLEX(G,k,N). Does G leav

least k a is N- all N. Proof:

all N-Plexe); fo

level i+1: ode U at level i: two cases :

Parent(U,i): closed unde exclusion, N- es are

smallest clique ial groups

LeftChild(U,i): eith mu U ∪ {v_ O(m+n) ves

-- RightChild(U,i): ode again

N < (|V|+2) / 2 then diam(G) ≤ 2

→ If an N-Plex has large “sloppyness” N “its”

& t ally realist

Homer

U understanding leaves

n) tim ds

at lea

(U – N(v_i+1)) ∪ {v_i+1}

leaves only … vel n

is maximal iff … NPLEX(G,k,N): "Does …

least k a" is NP-complete for all N. Proof: Informal argumer

CLIQUE(G,k) … PLEX(G,k,1); formal: reduce CLIQUE/C

two cases :

closed under exclusion

ial groups

{v_i+1} … O(m+n) … v_i+1]]

-- Right child(…,i):

N < (|V|+2) / 2 then diam(G) ≤ 2

→ If an N-Plex has a large "sloppyness" N "its"

& t … ally realist

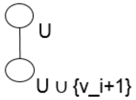
Hom…

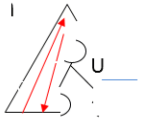
U under ful behaves

(G([U])) ≥ … logical

n) time

t lea





(U – N[v_i+1]) ∪ {v_i+1}

If U1 …

ery desirable

is maximal iff … NPLEX(G,k,N): "Does …

least k a" is NP-complete for all N. Proof: Informal argumer

CLIQUE(G,k) … PLEX(G,k,1); formal: reduce CLIQUE/C

two cases :

closed under exclusion

ial groups

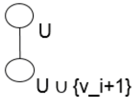
not r…

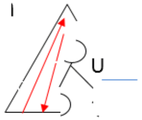
-- Right child(…,i):

U = V is adjacent to … o(G([U … v_i+1

→ If an N-Plex has a large "sloppyness" N "its"

II. N-cores under … behaves





(U – N[v_i+1]) ∪ {v_i+1}

If U1 …

ery desirable

is max … path: … maximum N-clique) … is (…)

least k a" is NP-complete for all N. Proof: Informal argumer

are connected maximal N-cores

ery desirable

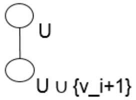
are … closed under exclusion (Example: cycle …)

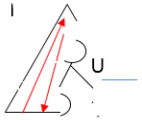
are generally not …

-- Right child(…,i):

→ If an N-Plex has a large "sloppyness" N "its"

II. N-cores under … behaves





(U – N[v_i+1]) ∪ {v_i+1}

If U1 …

ery desirable

is max … path: … maximum N-clique) … is (…)

least k a" is NP-complete for all N. Proof: Informal argumer

are connected maximal N-cores

are … closed under exclusion (Example: cycle …)

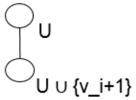
are generally not …

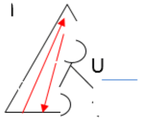
-- Right child(…,i):

→ If an N-Plex has a large "sloppyness" N "its" … groups

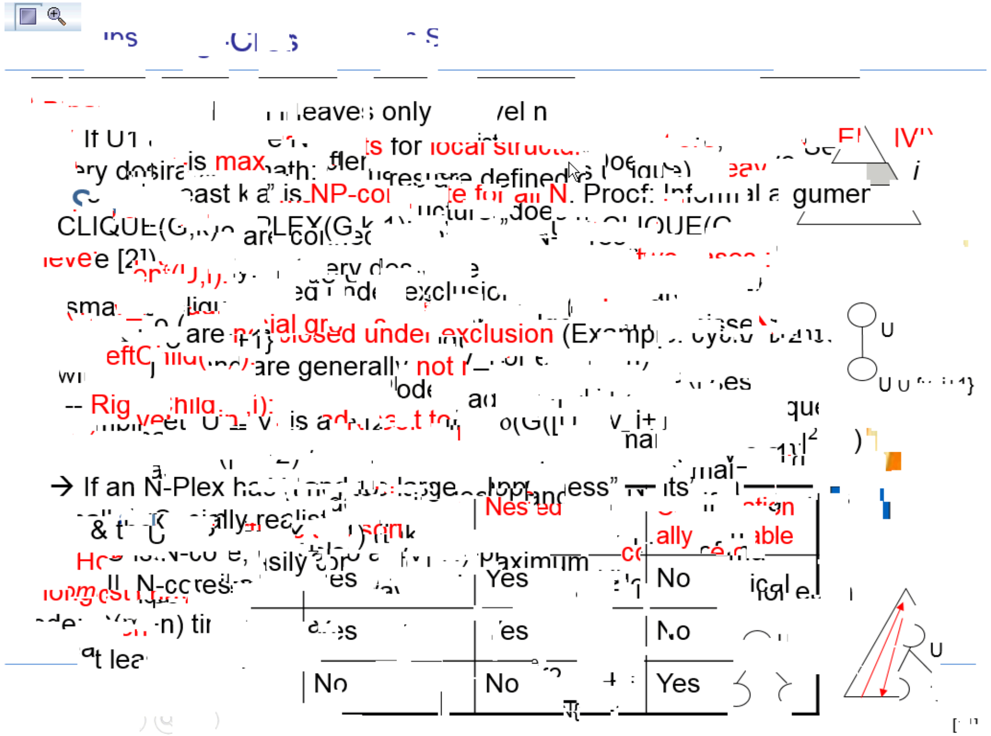
easily … maximum N-core

II. N-cores …





(U – N[v_i+1]) ∪ {v_i+1}



leaves only … vel n

If U1 … is max … fler … ts for local structu … define … (clique) … leav …

… is NP-co … te for all N. Proof: Informal argumer

CLIQUE(G,k) … PLEX(G,k) … structure „doe … CLIQUE(G

… are partial gr … closed under exclusion (Example: cycle …

… are generally not r …

-- Rig … hild … i):

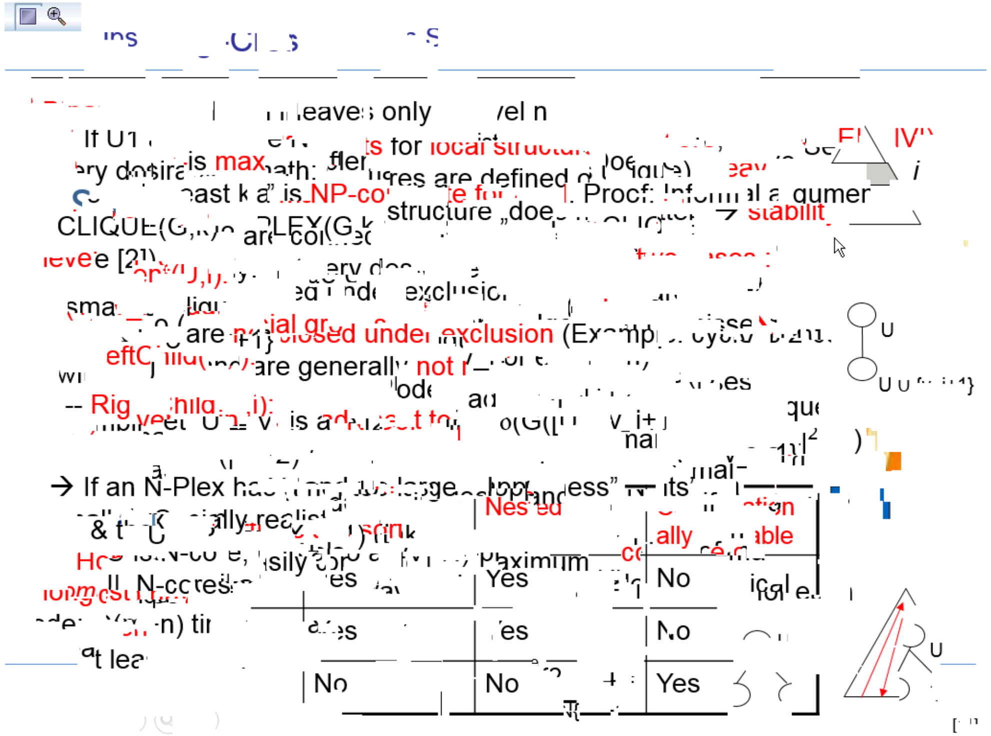
… V is a … t to … o(G([… V_i+1 … nal

→ If an N-Plex has … large … “ … ness” … its …

& t … ally realis …

	Nes ed		ally	able
	Yes		No	
	es		No	
No	No		Yes	

… t lea …



leaves only … vel n

If U1 … is max … fler … ts for local structu … res are defined … (clique) … leav …

… is NP-co … te for … Proof: Informal argumer

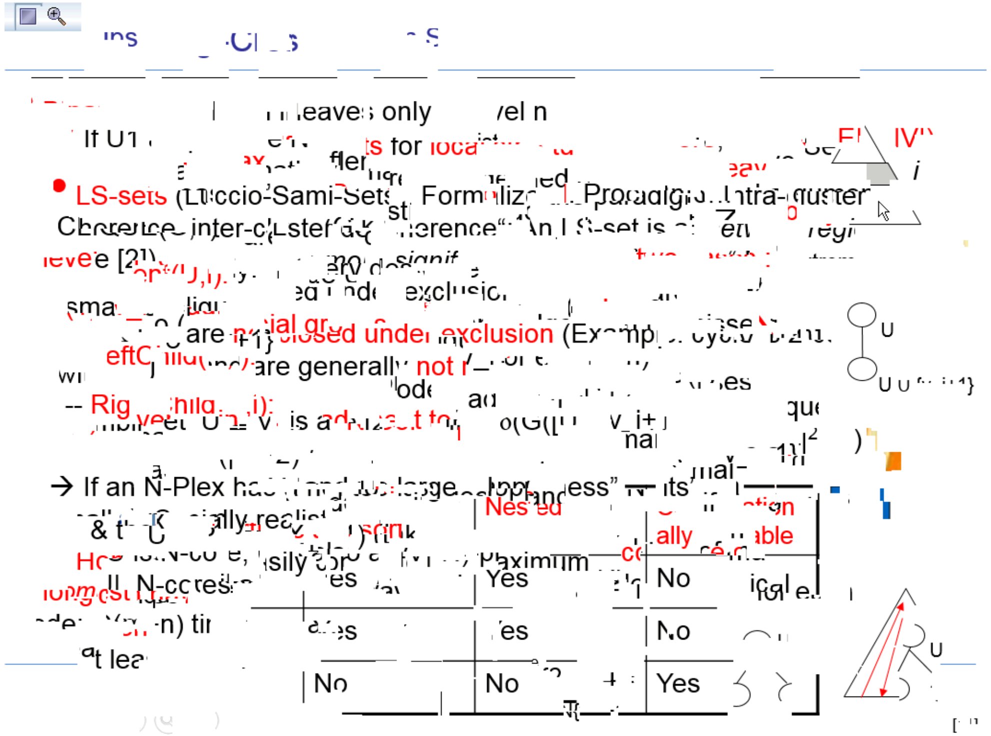
CLIQUE(G,k) … PLEX(G,k) … structure „doe … → stabilit

… are partial gr … closed under exclusion (Example: cycle …

… are generally not r …

→ If an N-Plex has … large … “ … ness” … its …

	Nes ed		ally	able
	Yes		No	
	es		No	
No	No		Yes	



leaves only … vel n

If U1 … ts for loca …

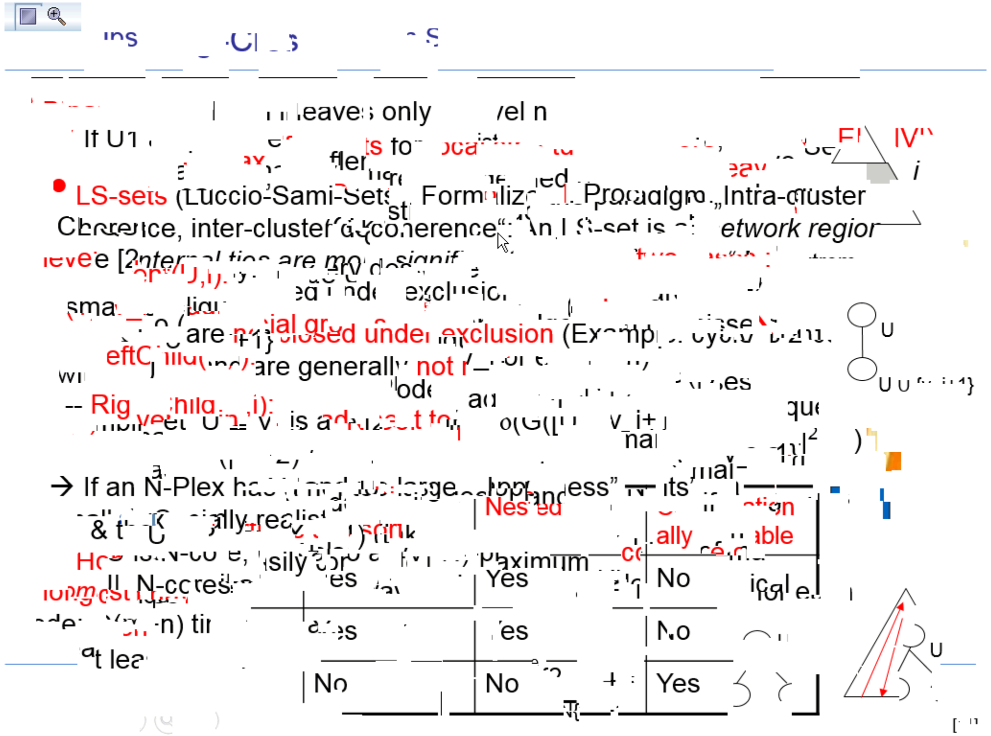
LS-sets (Luccio-Sami-Sets … Formalized … Proc algo … „Intra-cluster Coherence, inter-cluster … coherence“: An LS-set is … regi

… are partial gr … closed under exclusion (Example: cycle …

… are generally not r …

→ If an N-Plex has … large … “ … ness” … its …

	Nes ed		ally	able
	Yes		No	
	es		No	
No	No		Yes	



leaves only … vel n

If U1 … ts to … loca …

LS-sets (Luccio-Sami-Sets … Formalized … Proc algo … „Intra-cluster Coherence, inter-cluster coherence“: An LS-set is … etwork region

… Internal ties are mo … signif …

… are partial gr … closed under exclusion (Example: cycle …

… are generally not r …

→ If an N-Plex has … large … “ … ness” … its …

	Nes ed		ally	able
	Yes		No	
	es		No	
No	No		Yes	

leaves only ... vel n

If U1 ...

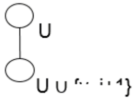
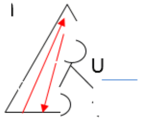
- LS-sets (Luccio-Sami-Sets): Formalized Paradigm „Intra-cluster Coherence, inter-cluster incoherence“: An LS-set is a „network region

... exclusion

... closed under exclusion ... are generally not ...

→ If an N-Plex has ... large ... less” ... its

	Nested	ally	able
	Yes	No	
	es	No	
No	No	Yes	

---

leaves only ... vel n

If U1 ...

- LS-sets (Luccio-Sami-Sets): Formalized Paradigm „Intra-cluster Coherence, inter-cluster incoherence“: An LS-set is a „network region

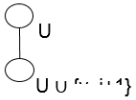
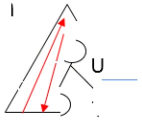
... exclusion

... closed under exclusion ... are generally not ...

ties with the network ...

→ If an N-Plex has ... large ... less” ... its

	Nested	ally	able
	Yes	No	
	es	No	
No	No	Yes	

---

leaves only ... vel n

If U1 ...

- LS-sets (Luccio-Sami-Sets): Formalized Paradigm „Intra-cluster Coherence, inter-cluster incoherence“: An LS-set is a „network region

... exclusion ... clique))

... closed under exclusion ... are generally not ...

ties with the network ...

U′⊂ U →

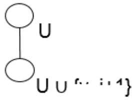
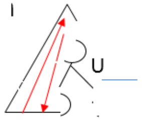
→ If an N-Plex ... me interesting properties ... no trival overlap

u1 ∩ u2 ≠ ∅ → u1⊆u2 or u2⊆u1; min degree ... LS ... is at least hal

number of ... may be good candidates ...

can ... a co...

	es	No
No	No	Yes

---

leaves only ... vel n

If U1 ...

Lambda-sets are ... Lambda

... connected to other members by more (edge disjoint) paths ... cluster

- LS-sets ... Coherence, inter-cluster incoherence ... is a „network region

... exclusion ... clique))

... closed under exclusion ... are generally not ...

ties with the network ...

U′⊂ U →

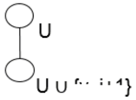
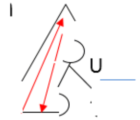
→ If an N-Plex ... me interesting properties ... no trival overlap

u1 ∩ u2 ≠ ∅ → u1⊆u2 or u2⊆u1; min degree ... LS ... is at least hal

number of ... may be good candidates ...

can ... a co...

	es	No
No	No	Yes y ( )

ıns ·Cl s S

leaves only vel n

If U1 ambda-sets are ts to ambda

- LS-sets connected to other members by more (edge disjoint) paths cluster

Coherence, inter-clu coherence it is a „*network region*"

level e [2*internal ties are more significant*] trem

λ(u,v) de excluded clique)) ies

sm v;

partial gr sed under exclusio

eft U ⊆ a Lambda set if proper

ties with t U does

min λ(u,v) > max λ(u,v)

u,v ∈ U u ∈U, v ∈ V-U

→ If an N-Plex I

ally re interesting properties no trival overlap

u1 ∩ u2 ≠ ∅ → u1⊆u2 or u2⊆u1; min degree LS is at least hal

number of N-cores may be good candid

longest (lo

order n) tin es No

at least

ca No No Yes y (