P/NP Notizen

Paper #49

18. Februar 2021

Aufbau von Paper #49

Titel: *A polynomial-Time Algorithm for the Maximum Clique Problem*
Author: *Zohreh O. Akbari*

I. Introduction

Eine wichtige Konsequenz des Cook-Levin Theorems (SAT Problem ist NP-vollständig) ist, dass sobald ein Problem aus NP in polynomieller Zeit lösbar ist, alle Probleme aus NP in polynomieller Zeit lösbar sind. Das würde bedeuten, dass $P = NP$.

Paper #49 präsentiert einen polynomiellen Algorithmus für das *Maximum Clique Problem*. Der Autor folgert also $P = NP$.

II. The Maximum Clique Problem

Definition 1 (Graph)**.**
Sei V eine Knotenmenge und $E \subseteq \{\{i,j\} \mid i,j \in V, i \neq j\}$. Dann heißt das Paar $G = (V,E)$ ein Graph.

Definition 2 (Vollständiger Graph)**.**
Ein Graph $G = (V,E)$ heißt vollständig, wenn alle Knoten paarweise adjazent sind. Also $\forall\ i,j \in V,\ i \neq j \Longrightarrow (i,j) \in E$. Falls jeder Knoten Knotengrad $|V|-1$ besitzt, so ist G vollständig.
Autor bezeichnet mit φ (vermutlich) den vollständigen Graphen.

Definition 3 (Clique)**.**
Sei $C \subseteq V$. C nennt man Clique, falls $G = (C, E')$ vollständig ist.

Definition 4 (Größte Clique)**.**
$\omega(G)$ ist die größte Clique in G. Also

$$\omega(G) = \max\{|S| : S \text{ ist eine Clique in } G\}$$

Definition 5 (Größter Teilgraph von G der α enthält)**.**
Sei α jener Knoten mit geringstem Knotengrad.

$$\alpha = \{j \in V : deg(j) \text{ ist minimal}\}$$

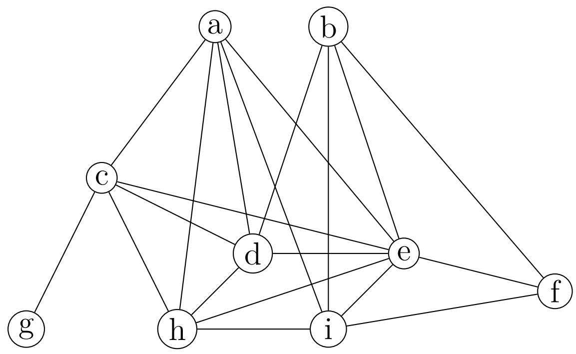
Den Teilgraph, der α und alle inzidenten Knoten zu α enhält, bezeichnet der Autor als größten Teilgraph in G.
(α und alle seine Nachbarn samt deren Kanten)

III. A Polynomial-Time Algorithm for the Maximum Clique Problem

Pseudocode:

```
MaxClique(G) {
    if (G is a complete graph)
        for each vertex of G: v
            if (|V| - 1 > max C[v])
                max C[v] := |V|;
                make max CP[v] point to a linked list containing V;
    else
    find the vertex of lowest degree: α
    find the largest subgraph of G in which α exists: G'(V',E')
    MaxClique(G');
    if (V - α≠φ)
        MaxClique(G - α);
}
```

Beispiel:



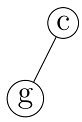
Graph G

Ausganglage ist der obige beliebige Graph G.

`1,7:` G ist nicht vollständig, da nicht jeder Knoten Knotengrad $|V|-1$ besitzt.

`8-9:` Knoten mit minimalen Knotengrad ist g, also $\alpha := g$.

$G' = (\{c,g\},\{\{c,g\}\})$

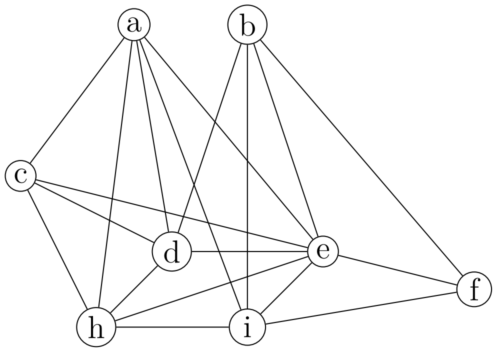


`10,2-6:` G' ist vollständig.

	a	b	c	d	e	f	g	h	i
maxC			2				2		
maxC									

c
g

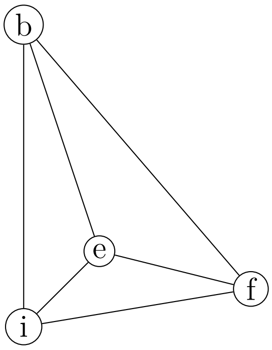
`11:` $V - g$ ist nicht vollständig.



$V - g$

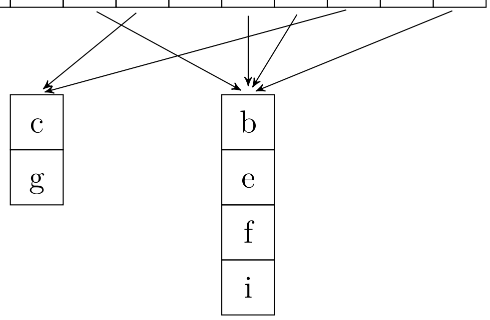
`12,8,9:` Knoten mit minimalen Knotengrad ist f, also $\alpha := f$.

$G' = (\{b,e,f,i\},\{\{b,e\},\{b,f\},\{b,i\},\{e,f\},\{e,i\},\{f,i\}\})$

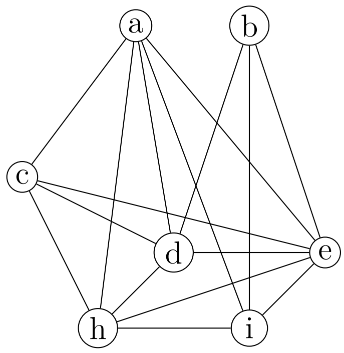


`10,2-6:` G' ist vollständig.

	a	b	c	d	e	f	g	h	i
maxC		4	2		4	4	2		4
maxC									

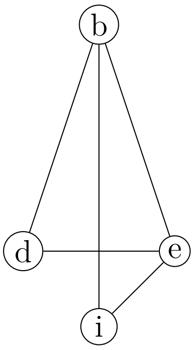


`11:` $V - f$ ist nicht vollständig.



$V - f$

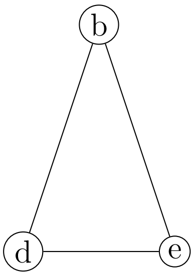
`12,8,9:` Knoten mit minimalen Knotengrad ist b, also $\alpha := b$.
$G' = (\{b,d,e,i\},\{\{b,e\},\{b,d\},\{b,i\},\{d,e\},\{e,i\}\})$



`10:` G' ist nicht vollständig.

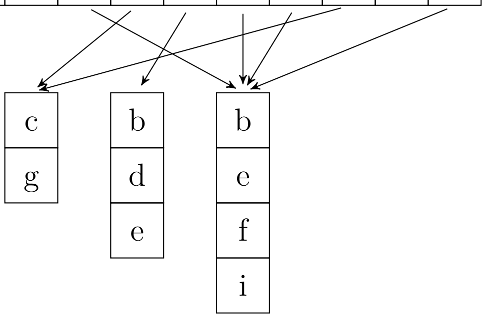
`8-9:` Knoten mit minimalen Knotengrad ist d oder i, also o.B.d.A? $\alpha := d$.

$G' = (\{b,d,e\},\{\{b,e\},\{b,d\},\{d,e\}\})$

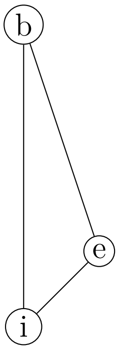


`10,2-6:` G' ist vollständig.

	a	b	c	d	e	f	g	h	i
maxC		4	2	3	4	4	2		4
maxC									

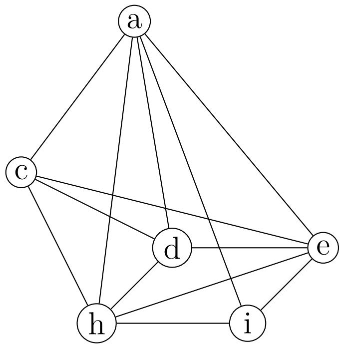


`11:` $G' - d$ ist vollständig. $(G' - d) = (\{b,e,i\},\{\{b,e\},\{b,i\},\{e,i\}\})$.



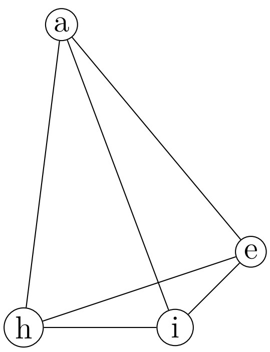
Linked List bleibt gleich.

`11:` $V - b$ ist nicht vollständig.



$V - b$

`12,8,9:` Knoten mit minimalen Knotengrad ist i, also $\alpha := i$.
$G' = (\{a,e,i,h\},\{\{a,e\},\{a,i\},\{a,h\},\{e,i\},\{e,h\},\{i,h\}\})$

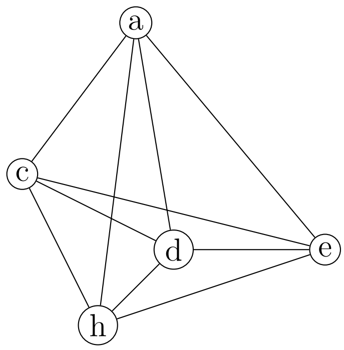


`10,2-6:` G' ist vollständig.

	a	b	c	d	e	f	g	h	i
maxC	4	4	2	3	4	4	2	4	4
maxC									

c	b	b	a
g	d	e	e
	e	f	i
		i	h

`11:` $V - i$ ist vollständig.



$V - i$

	a	b	c	d	e	f	g	h	i
maxC	4	4	2	3	4	4	2	4	4
maxC									

c	b	b	a
g	d	e	e
	e	f	i
		i	h

Problem bei Codezeile 4:

`if (|V| - 1 > max C[v])`

`C[a] = 4, |V| - 1 = 4, 4` $\not> 4$

Die If Bedingung ist hier nicht erfüllt, somit würde der Algorithmus die Zeilen `4-5` überspringen. Der Autor fürt jedoch in diesem Beispiel diese Zeilen aus.

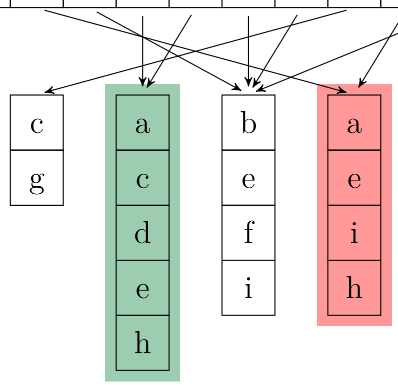
Annahme: Codezeile `4` müsste folgendermaßen geändert werden:

`if (|V| > max C[v])`

Man erhält in beiden Fällen die größte Clique mit 5 Knoten. Die Liste bzw. verlinkte Liste ist jedoch eine andere. Diese ist nicht korrekt.

Ergebnis ohne Änderung:

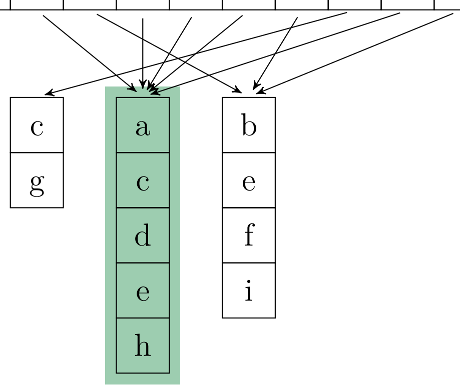
	a	b	c	d	e	f	g	h	i
maxC	4	4	5	5	4	4	2	4	4
maxC									



Maximale Clique enthält 5 Knoten: $\{a, c, d, e, h\}$

Ergebnis mit Änderung:

	a	b	c	d	e	f	g	h	i
maxC	5	4	5	5	5	4	2	5	4
maxC									



Maximale Clique enthält 5 Knoten: $\{a, c, d, e, h\}$