ETH zürich
Departement Informatik

Algorithmik für schwere Probleme

Prof. Dr. Dennis Komm
Dr. Hans-Joachim Böckenhauer
Dr. Richard Kralovič

`https://courses.algodi.inf.ethz.ch/schwereprob-2025`

Lösungsvorschläge – Blatt 3

Zürich, 11. März 2025

Lösung zu Aufgabe 3

(a) Der modifizierte Algorithmus ist wie folgt definiert:

Eingabe: Ein Graph $G = (V, E)$ mit n Knoten.
Ausgabe: Alle inklusions-maximalen Independent-Sets S von G.

function *All-IS*(G) :
  if $V = \emptyset$ **then return** $\{\emptyset\}$;
  else
    $v \leftarrow \{v \in V \mid \delta(v) = \delta_{\min}(G)\}$;
    return $\bigcup_{u \in N[v]} \{\{u\} \cup S \mid S \in \text{All-IS}(G - N[u])\}$;
  end

Wir zeigen nun die Korrektheit des Algorithmus per Induktion über die Anzahl Knoten n. Im Fall $n = 0$ ist die leere Menge das einzige Independent-Set des leeren Graphen. Also ist $\text{All-IS}(G) = \{\emptyset\}$ korrekt. Falls $n > 0$, dann wählt der Algorithmus $\text{All-IS}(G)$ einen Knoten v mit minimalem Knotengrad $\delta(v) = \delta_{\min}(G)$ und gibt die Mengen

$$\bigcup_{u \in N[v]} \{\{u\} \cup S \mid S \in \text{All-IS}(G - N[u])\}$$

zurück. Aus der Induktionsannahme folgt für jedes $u \in N[v]$, dass $\text{All-IS}(G - N[u])$ alle inklusions-maximalen Independent-Sets von $G - N[u]$ zurückgibt. Sei $S \in \text{All-IS}(G - N[u])$. Dann ist $\{u\} \cup S$ offenbar ein Independent-Set von G. Die Menge $\{u\} \cup S$ lässt sich offenbar durch keinen Nachbarn $w \in N(u)$ von u zu einen Independent-Set von G erweitern. Falls sich die Menge $\{u\} \cup S$ durch einen Knoten $w \in V(G) \setminus N[u] \setminus S$ zu einem Independent-Set $\{u\} \cup (\{w\} \cup S)$ von G erweitern liesse, dann liesse sich auch die Menge S zu einem Independent-Set $\{w\} \cup S$ von $G - N[u]$ erweitern, was zu einem Widerspruch führt. Deshalb ist die Menge $\{u\} \cup S$ ein inklusions-maximales Independent-Set von G. Es folgt, dass $\text{All-IS}(G)$ nur inklusions-maximale Independent-Sets von G zurückgibt.

Sei nun S ein beliebiges inklusions-maximales Independent-Set von G. Wir zeigen, dass $S \in \text{All-IS}(G)$, also dass $\text{All-IS}(G)$ alle inklusions-maximalen Independent-Sets von G zurückgibt. Falls S weder den Knoten v noch einen Nachbarn $u \in N(v)$

enthält, dann lässt sich S durch den Knoten v zu einem Independent-Set $\{v\} \cup S$ von G erweitern, was zu einem Widerspruch führt. Sei $u \in N[v]$, also entweder $u = v$ oder $u \in N(v)$, wobei $u \in S$ ein beliebiger Knoten in S ist. Dann kann das Independent-Set S keinen Knoten aus $N[u]$ enthalten und die Menge $S \setminus \{u\}$ ist ein Independent-Set von $G - N[u]$. Es reicht zu zeigen, dass $S \setminus \{u\} \in \textsc{All-IS}(G - N[u])$, also dass $S \setminus \{u\}$ inklusions-maximal bezüglich $G - N[u]$ ist. Falls sich $S \setminus \{u\}$ durch einen Knoten $w \in V(G) \setminus N[u] \setminus S$ zu einem Independent-Set von $G - N[u]$ erweitern liesse, dann liesse sich auch die Menge $S = \{u\} \cup (S \setminus \{u\})$ zu einem Independent-Set $S \cup \{w\} = \{u\} \cup (\{w\} \cup (S \setminus \{u\}))$ von G erweitern, was zu einem Widerspruch führt. Aus $S \setminus \{u\} \in \textsc{All-IS}(G - N[u])$ folgt schliesslich, dass $S = \{u\} \cup (S \setminus \{u\}) \in \textsc{All-IS}(G)$.

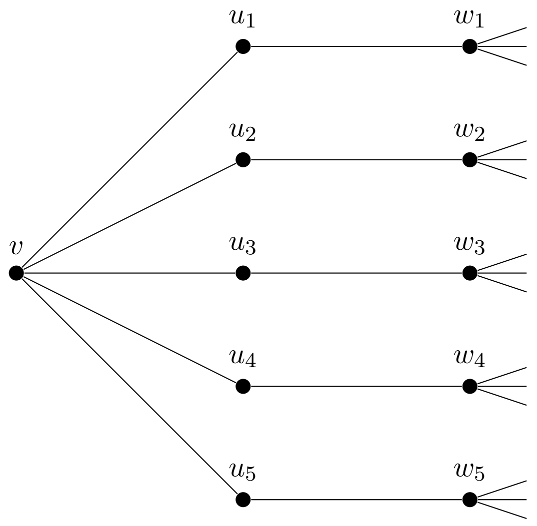
(b) Der Algorithmus führt zur selben Baumstruktur wie der Algorithmus aus der Vorlesung, der ein maximales Independent-Set berechnet. In der Vorlesung wurde gezeigt, dass dieser Baum höchstens $3^{n/3}$ Blätter hat. Da jedes solche Blatt nur einem inklusions-maximalen Independent Set entspricht und der Algorithmus alle diese generiert, kann es auch höchstens $3^{n/3}$ inklusions-maximale Independent-Sets geben.

(c) Sei $n \in \mathbb{N}$ eine beliebige durch 3 teilbare natürliche Zahl. Wir betrachten den Graphen T_n auf n Knoten, der aus $n/3$ paarweise disjunkten Dreiecken besteht. Ein inklusions-maximales Independent-Set von T_n besteht aus genau einem Knoten von jedem der $n/3$ Dreiecke. Insgesamt gibt es also $3^{n/3} \approx 1.442^n$ inklusions-maximale Independent-Sets von T_n und die Abschätzung aus Teilaufgabe (b) lässt sich deshalb nicht mehr verbessern. Damit ist auch klar, dass kein Algorithmus eine bessere Laufzeit haben kann.

Lösung zu Aufgabe 4

In dieser Aufgabe betrachten wir das Mass $w(G) = n_{\geq 3}$, wobei $n_{\geq 3}$ die Anzahl Knoten mit Grad mindestens 3 bezeichnet. Der Algorithmus aus der Vorlesung verzweigt in die Teilgraphen $G_{\text{out}} = G - v$ und $G_{\text{in}} = G - N[v]$, wobei v einen Knoten von maximalem Grad $\delta(v) = \delta_{\max}(G)$ in einem Graphen G bezeichnet, der keine Knoten von Grad höchstens 1 enthält (der Algorithmus macht ohne Verzweigung mit der einzigen Instanz $G - N[v]$ weiter, solange es einen Knoten v mit $\delta(v) \leq 1$ in G gibt).

Wir untersuchen nun den folgenden Fall: der Grad vom Knoten v beträgt mindestens 4, $\delta(v) \geq 4$, alle Nachbarn $u \in N(v)$ vom Knoten v haben Grad $\delta(u) = 2$ und alle Nachbarn $w \in N(u)$ von Knoten $u \in N(v)$ haben Grad mindestens 4, $\delta(w) \geq 4$. Eine mögliche Umgebung vom Knoten v ist auf der folgenden Abbildung dargestellt.



Wir berechnen nun $w(G_{\text{out}}) = w(G_{\text{in}}) = w(G) - 1$, da die Anzahl Knoten mit Grad mindestens 3 in G_{out} und G_{in} jeweils um 1 sinkt (wegen dem Knoten v mit $\delta(v) \geq 4$). Die Knoten w_i haben nämlich auch nach Entfernung der Knoten u_i in G_{in} Grad mindestens 3 und deshalb die Anzahl Knoten mit Grad mindestens 3 nicht beeinflussen.

In diesem Fall ergibt sich also ein Branching-Vector von $(1, 1)$ mit dem charakteristischen Polynom $x^n - 2 \cdot x^{n-1} = 0$. Die einzige positive Nullstelle von diesem Polynom ist $x = 2$ und der Branching-Factor beträgt somit $\alpha = 2$. Die Laufzeit von dem Algorithmus lässt sich also lediglich durch $\mathcal{O}^*(2^n)$ abschätzen. Die vereinfachte Analyse mit dem Mass $w(G) = n_{\geq 3}$ bringt also keine Verbesserung bezüglich dem naiven Algorithmus, der alle Teilmengen der Knotenmenge in $\mathcal{O}^*(2^n)$ durchiteriert.