Prof. Dr. A. Goerdt
Julian Pape-Lange

TU Chemnitz
Wintersemester 2019/2020
29.10.2019

Theoretische Informatik I

3. Übung

Abgabe: Lösen Sie Aufgabe **4**. Ihre Lösungen geben Sie bitte entweder

- vor oder nach der Vorlesung am 05.11.2019 oder
- bis zum 05.11.2019 um 13:00 Uhr per Mail
  an `julian.pape-lange@informatik.tu-chemnitz.de`
  mit *Betreff:* `TI1 Hausaufgaben`

ab.

1. Aufgabe:

(a) Sei $c > 1$ eine beliebige Konstante. Zeigen Sie, dass es ein $n_0 = n_0(c)$ gibt, so dass für alle $n > n_0$ gilt:

$$2^n > n^c$$

(b) Folgern Sie aus a), dass für jede noch so große Konstante k und jede noch so kleine Konstante $d > 1$

(i) $d^n > n^k$ (ii) $n > (\ln n)^k$ (iii) $d^{\sqrt[k]{n}} > n^k$

gilt, wenn $n > n_0 = n_0(d, k)$ erfüllt ist.

2. Aufgabe: Sei $G = (V, E)$ ein ungerichteter Graph mit mindestens zwei Knoten. Zeigen Sie folgende Aussagen:

(a) Die Summe der Grade aller Knoten ist durch 2 teilbar.

(b) Es gibt stets zwei Knoten, die denselben Grad haben.

3. Aufgabe: Ein ungerichteter Graph $G = (V, E)$ wird *bipartit* oder *2-färbbar* genannt, wenn es zwei Mengen V_1 und V_2 gibt, so dass folgendes gilt:

- $V = V_1 \cup V_2$ und $V_1 \cap V_2 = \emptyset$.
- Für alle Kanten $\{u, v\} \in E$ gilt entweder $u \in V_1$ und $v \in V_2$ oder $u \in V_2$ und $v \in V_1$. Das heißt, es gibt keine Kante, bei der beide Endpunkte zu V_1 oder beide Endpunkte zu V_2 gehören.

Der Begriff *2-färbbar* wird benutzt, weil man die Knoten des Graphen so mit zwei Farben färben kann, dass keine Kante zwei gleich gefärbte Knoten verbindet.

(a) Geben Sie einen Algorithmus an, der mit einer Laufzeit $O(|V| + |E|)$ erkennt, ob ein gegebener ungerichteter Graph G *2-färbbar* ist. Verwenden Sie die Breitensuche und die von der Breitensuche gelieferten Distanzwerte.

(b) Analog zu 2-färbbaren Graphen sind *3-färbbare* Graphen definiert. Versuchen Sie, Ihre Idee aus (a) für einen Test auf 3-Färbbarkeit zu übertragen.

(c) Geben Sie einen Algorithmus an, der erkennt, ob ein gegebener Graph 3-färbbar ist oder nicht. Bestimmen Sie die Laufzeit Ihres Verfahrens.

4. Aufgabe: Betrachten Sie folgenden Algorithmus, der als Eingabe einen ungerichteten Graphen $G = (V, E)$ erhält.

1. $G' = G$
2. `Solange es in` G' `einen Knoten` u `mit Grad(`u`)` ≤ 2 `gibt:`
   `Entferne` u `(mit all seinen Kanten) aus` G'
3. `Falls` G' `leer ist: Ausgabe '`G `ist 3-färbbar.'`
4. `Sonst: Ausgabe 'Weiß nicht.'`

(a) Wir betrachten den Graphen $G = (V, E)$ mit

$$
\begin{aligned}
V &= \big\{1, 2, 3, 4, 5, 6, 7\big\} \\
E &= \big\{\{1, 2\}, \{2, 3\}, \{1, 4\}, \{4, 3\}, \{2, 4\}, \{1, 5\}, \{6, 5\}, \{7, 5\}, \{5, 4\}\big\}.
\end{aligned}
$$

Geben Sie G' *nach jedem Durchlauf* von Schritt 2 an.

(b) Geben Sie einen *3-färbbaren* (siehe Aufgabe 3) Graphen G an, bei dem obiger Algorithmus die Antwort „weiß nicht“ liefert. Geben Sie auch G' am Ende von Schritt 2 und eine gültige 3-Färbung Ihres Graphen an!

(c) Zeigen Sie die Korrektheit des Algorithmus. Das heißt: Wenn der Graph *nicht* 3-färbbar ist, darf auf keinen Fall „G ist 3-färbbar.“ ausgegeben werden!

Hinweis: Verfolgen Sie den Algorithmus vom Ende her zum Anfang hin. ($\rightarrow$ Induktionsbeweis)