Algorithmen und Datenstrukturen

Blatt 5

Uni Hamburg, Wintersemester 2019/20

Präsentation am 08.–10. Januar 2020

Jede Teilaufgabe zählt als ein einzelnes Kreuzchen.

Übung 1.

Beweisen oder widerlegen Sie folgende Aussagen zu ungerichteten Graphen:

(a) Es gilt für $G = (V, E)$:

$$\sum_{v \in V} deg(v) = 2|E|$$

(b) Seien v, w die einzigen beiden Knoten in $G = (V, E)$ mit ungeradem Grad, so sind v und w über einen Pfad in G verbunden.
Hinweis: Nutzen Sie die Aussage aus Aufgabe a).

Lösung 1.

(a) Sei m die Anzahl der Kanten.
Induktionsvoraussetzung: Für Graphen mit höchstens m Kanten gilt:

$$\sum_{v \in V} deg(v) = 2m.$$

Induktionsanfang: Ein Graph ohne Kanten ($m = 0$) hat nur Knoten mit Grad 0.
Induktionsschluss: Angenommen der Graph G hat $m + 1$ Kanten. Entferne eine zufällige Kante. Der daraus resultierende Graph $G' = (V, E')$ hat m Kanten. Es gilt die IV für G': $\sum_{v \in V} deg(v) = 2m$. G hat genau die entfernte Kante zusätzlich. Diese verbindet genau zwei Knoten in G, wodurch deren Grad jeweils um 1 erhöht wird. Daher gilt für G: $\sum_{v \in V} deg(v) = 2m + 2 = 2(m + 1)$.

(b) Angenommen v und w sind nicht verbunden. Dann lässt sich V in zwei Mengen V_1 und V_2 partitionieren, sodass kein Knoten aus V_1 mit einem Knoten aus V_2 verbunden ist. Das heißt, $G = (V_1 \uplus V_2, E_1 \uplus E_2)$[1]. Dann ist $G_1 = (V_1, E_1)$ ein Graph.
Sei o.B.d.A. $v \in V_1, w \notin V_1$. Dann gilt $\forall x \in V_1 \setminus \{v\} : deg(x)$ ist gerade. Damit ist $\sum_{x \in V_1} deg(x)$ ungerade und nach Aufgabe 1.a) G_1 kein Graph.
Durch diesen Widerspruch schließen wir, dass v und w verbunden sind.

[1] $A \uplus B$ bezeichnet die disjunkte Vereinigung, d.h. $A \cap B = \emptyset$.

Übung 2.
Ein ungerichteter Graph $G = (V, E)$ ist bipartit, falls zwei Mengen $U, W \subseteq V$ existieren, sodass

- $U \cup W = V$
- $U \cap W = \emptyset$
- $E \subseteq \{\{v_1, v_2\} : v_1 \in U \wedge v_2 \in W\}$, d.h. Knoten in U bzw. W sind nicht untereinander adjazent.

(a) Zeigen Sie: G ist bipartit $\Leftrightarrow$ G enthält keine Zyklen ungerader Länge $k \geq 3$.

(b) Geben Sie einen möglichst effizienten Algorithmus an, der für einen zusammenhängenden Graphen G entscheidet, ob dieser bipartit ist.

(c) Begründen Sie die Korrektheit und Laufzeit ihres Algorithmus.
Hinweis: Nutzen Sie die Aussage aus Aufgabe a).

Lösung 2.

(a) ($\Rightarrow$) Seien G bipartit und $U, W \subseteq V$ die entsprechenden disjunkten Knotenmengen(siehe Aufgabenstellung). Angenommen, G enthält einen Zyklus C ungerader Länge $k \geq 3$. Sei $C = (v, v_1, \ldots, v_{2j}, v)$ ein solcher Zyklus für $j \geq 1$. O.B.d.A. sei $v \in U$. Dann muss v_1 in W enthalten sein aufgrund der Bipartitheit. Induktiv muss demnach $v_{2j} \in U$ und daher $v \in W$ gelten, was zu einem Widerspruch der Annahme führt.

($\Leftarrow$) Sei G ein Graph,welcher keine Zyklen ungerader Länge $k \geq 3$ enthält. O.B.d.A sei G zusammenhängend (da wir Zusammenhangskomponenten getrennt voneinander untersuchen können). Seien $v \in V$,

$$X = \{x \in V | \text{ der kürzeste Pfad zwischen } x \text{ und } v \text{ hat gerade Länge}\}$$

und $Y = V \setminus X$. Seien nun $x, x' \in X$, $x \neq x'$ und nehme an, dass $\{x, x'\} \in E$. Wenn $x = v$, dann hat der kürzeste Pfad von x' nach v ungerade Länge (1), daher gilt $x \neq v$ und analog auch $x' \neq v$. Dann existiert ein Zyklus $\{v, \ldots, x, x', \ldots, v\}$ ungerader Länge, was ein Widerspruch ist. Demnach sind keine zwei Knoten aus X adjazent zueinander und analog dazu gilt das auch für zwei Knoten aus Y. Damit ist G bipartit mit X und Y.

(b) Der folgende Algorithmus entscheidet ob ein Graph G bipartit ist.

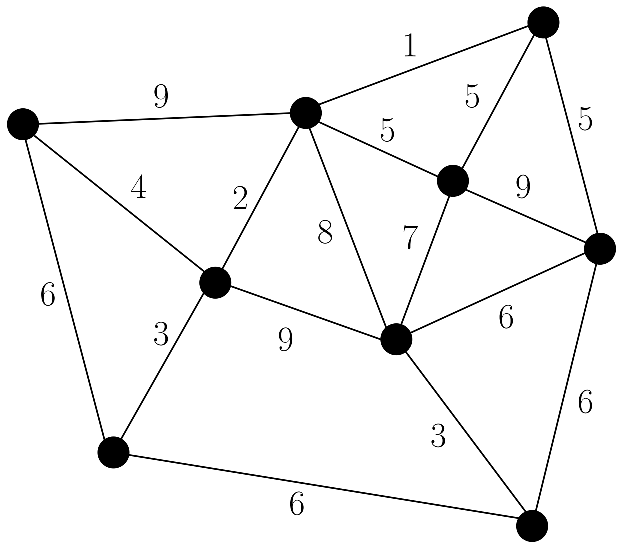
```
BipartiteTest(G)
s = any node of G
for each u ∈ G.V \ {s}
    d[u] = ∞
d[s] = 0
Q = {s}
while Q ≠ ∅
    u = Dequeue(Q)
    for each v ∈ u.Adj
        if d[v] == ∞
            Enqueue(Q, v)
            d[v] = d[u] + 1
        else if d[v] == d[u]
            return NOT BIPARTITE
return G IS BIPARTITE
```

(c) Der Algorithmus beruht auf der Breitensuche und hat daher Laufzeit $O(|V|+|E|)$. Der Algorithmus testet, ob ein Zyklus ungerader Länge C in dem Graphen vorhanden ist. Nach Aufgabe a) ist dies äquivalent dazu, dass der Graph bipartit ist. Sei w der vom Algorithmus gewählte Knoten mit $d[w] = 0$. Angenommen der Algorithmus findet zwei Knoten u, v, sodass $d[u] = d[v]$ und $\{u, v\} \in E$. Dann existiert ein Zyklus ungerader Länge: $(w, \ldots, u, v, \ldots, w)$.
Andersherum nehmen wir an, dass keine zwei Knoten u, v existieren, sodass $d[u] = d[v]$ und $\{u, v\} \in E$. Dann verlaufen alle Kanten zwischen Knoten mit $d[u] = d[v] + 1$. Ein Zyklus hat daher immer gerade Länge.

Übung 3.
Sei $G = (V, E)$ ein zusammenhängender, gewichteter Graph mit Gewichtsfunktion $w : E \to \mathbb{R}$. Ein Spannbaum ist ein Baum $T = (V, E')$ mit $E' \subseteq E$, der aus einer Teilmenge der Kanten (und allen Knoten) von G besteht, sodass alle Knoten in V miteinander verbunden sind. Wir betrachten zwei Varianten von Spannbäumen:

- Ein minimaler Spannbaum ist ein Spannbaum, dessen Gesamtgewicht $w(T) = \sum_{(u,v)\in E'} w(u, v)$ unter allen Spannbäumen von G minimal ist.
- Ein minimaler Bottleneck-Spannbaum ist ein Spannbaum, bei den das maximale Kantengewicht unter allen Kanten von T minimal ist (im Gegensatz zur Summe der Kantengewichte beim minimalen Spannbaum). Bei einem minimalen Bottleneck-Spannbaum ist also $\max\{w(u, v) \mid (u, v) \in E'\}$ minimal.

(a) Zeichnen Sie jeweils einen minimalen Spannbaum und einen minimalen Bottleneck-Spannbaum in folgenden Graphen ein:



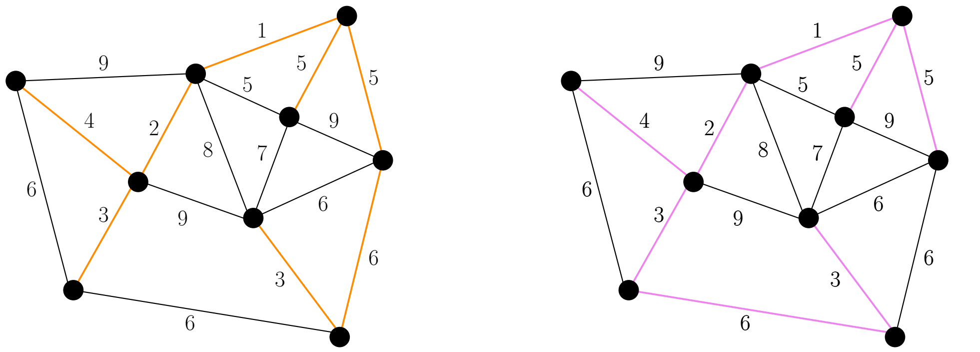
(b) Zeigen Sie: Jeder minimale Spannbaum eines Graphen G ist auch ein minimaler Bottleneck-Spannbaum von G.

(c) Widerlegen Sie durch Angabe eines minimalen Gegenbeispiels: Jeder minimale Bottleneck-Spannbaum eines Graphen G ist auch ein minimaler Spannbaum von G.

Lösung 3.

(a) Offensichtlich ist aufgrund von b) auch der minimale Spannbaum ein minimaler Bottleneck-Spannbaum. Wir haben trotzdem noch einen alternativen minimalen Bottleneck-Spannbaum angegeben.

(b) Sei T ein minimaler Spannbaum von G. Wir nehmen an, dass T kein minimaler Bottleneck-Spannbaum ist. Des Weiteren nehmen wir an, dass T' ein minimaler Bottleneck-Spannbaum ist mit $b = \max\{w(x, y) \mid (x, y) \in E_{T'}\}$ und es existiert



(a) Minimaler Spannbaum (b) Minimaler Bottleneck-Spannbaum

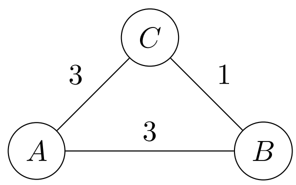
Abbildung 1: Beispiel

eine Kante $\{u, v\} \in E_T$ mit $w(u, v) > b$. Wir betrachten nun die folgenden Knotenmengen A, B von T:

$$A = \{u' \in V_T \mid \exists \text{ Pfad in } T \text{ zwischen } u' \text{ und } u, \text{ welcher nicht } v \text{ enthält}\}$$
$$B = \{v' \in V_T \mid \exists \text{ Pfad in } T \text{ zwischen } v' \text{ und } v, \text{ welcher nicht } u \text{ enthält}\}$$

Das Tupel (A, B) bildet einen Schnitt über T, d.h. die beiden Knotenmengen A, B bilden eine Partition über die Knoten in T. Da T' auch ein Spannbaum ist, muss eine Kante $\{x, y\} \in E_{T'}$ existieren, so dass $x \in A$ und $y \in B$ gelte. Zudem wissen wir aus unserer Annahme, dass $w(x, y) < w(u, v)$ gelte.
Nun betrachten wir den folgenden Baum $T'' = (V_T, E_T \setminus \{u, v\} \cup \{x, y\})$. Wir stellen fest, dass es sich um einen Spannbaum handelt, da wir nur eine Kante zwischen A und B ausgetauscht haben. Des Weiteren gilt nun $w(T'') < w(T)$, wegen $w(x, y) < w(u, v)$, was wiederum unsere Annahme widerspricht, dass T ein minimaler Spannbaum gewesen ist. Demnach ist T auch ein minimaler Bottleneck-Spannbaum.

(c) Betrachte den folgenden Graph G:



Es ist einfach zu sehen, dass $\{(a, b), (a, c)\}$ ein minimaler Bottleneck-Spannbaum ist, aber kein minimaler Spannbaum.