Technische Universität München
Zentrum Mathematik
Prof. Dr. P. Gritzmann, Dipl.-Inf. Dipl.-Math. S. Borgwardt, Dipl.-Math. M. Ritter

Optimierung 2, WS 2008/09

Übungsblatt 4

Im Folgenden sei ein Graph G gegeben durch:

- eine Knotenmenge $V = \{v_1, ..., v_n\}$,
- eine Funktion $N : V \to 2^V$, die für jeden Knoten $v \in V$ die Menge der zu v benachbarten Knoten $N(v)$ liefert (*Adjazenzliste*).

Beachten Sie, dass G ein ungerichteter Graph ist, N ist also symmetrisch in dem Sinne, dass $w \in N(v) \Leftrightarrow v \in N(w)$ für zwei Knoten $v, w \in V$ gilt.

Betrachten Sie für die folgenden Aufgaben jeweils zunächst den folgenden Graphen G, gegeben durch:

- $V = \{v_1, ..., v_7\}$;
- $N(v_1) = \{v_2, v_4, v_6\}$, $N(v_2) = \{v_1, v_4\}$, $N(v_3) = \{v_5\}$, $N(v_4) = \{v_1, v_2\}$, $N(v_5) = \{v_3\}$, $N(v_6) = \{v_1, v_7\}$, $N(v_7) = \{v_6\}$.

Formulieren Sie dann Ihre Ergebnisse auch für beliebige Graphen, die durch ihre Adjazenzliste gegeben sind.

Aufgabe 4.1
Sei G ein Graph, der durch die Knotenmenge V und eine Adjazenzliste N gegeben ist. Es stehen Ihnen folgende Datenstrukturen und Operationen zur Verfügung:

- Ein Stack S, auf dem Sie einen Knoten oben ablegen können, und von dem Sie den obersten Knoten wegnehmen können (Last In - First Out).
- Das Markieren eines Knotens und ein Test, ob ein Knoten markiert ist.
- Das Auflisten aller Nachbarn eines Knotens (mit der Adjazenzliste N).

Entwickeln Sie einen Algorithmus, der bestimmt, ob G zusammenhängend ist. Begründen Sie die Korrektheit Ihres Algorithmus.

Lösung zu Aufgabe 4.1
Gesucht ist der Tiefensuche-Algorithmus:

- 1. Wähle einen beliebigen Knoten, lege ihn auf den Stack S und markiere ihn. Initialisiere eine Zählvariable $k = 0$.

- 2. Lege alle nicht-markierten Nachbarn des obersten Knotens auf S und markiere sie. Falls kein solcher nicht-markierter Nachbar des obersten Knotens existiert, nimm den obersten Knoten vom Stack und erhöhe k um 1. Falls der Stack jetzt nicht leer ist, wiederhole 2.
- Falls $k < |V|$, ist G nicht zusammenhängend, sonst ($k = |V|$) ist G zusammenhängend.

Alle Knoten beginnen unmarkiert. Wir betrachten die Zusammenhangskomponente von v. Jeder Knoten darin wird irgendwann während des Algorithmus genau einmal (als Nachbar eines markierten Knotens) entdeckt und markiert auf den Stack gelegt. Spätestens wenn alle Knoten der Zusammenhangskomponente markiert sind, sind auch alle Nachbarn des obersten Knotens markiert, und er wird abgearbeitet. Hierdurch wird jeder Knoten der Zusammenhangskomponente einmal zu einem obersten Knoten, dessen Nachbarn alle markiert sind, und trägt beim Entfernen aus dem Stack also zu einer 1 der Zählvariable k bei. Wir arbeiten bis der Stack leer ist, womit k genau die Anzahl der Knoten in der Zusammenhangskomponente von v repräsentiert.

Aufgabe 4.2
Sei G ein Graph, der durch die Knotenmenge V und eine Adjazenzliste N gegeben ist. Es stehen Ihnen folgende Datenstrukturen und Operationen zur Verfügung:

- Eine Warteschlange Q, an deren Ende Sie einen Knoten anhängen können, und von deren Anfang Sie einen Knoten herausnehmen können (First In - First Out).
- Das Markieren eines Knotens und ein Test, ob ein Knoten markiert ist.
- Das Auflisten aller Nachbarn eines Knotens (mit der Adjazenzliste N).

Entwickeln Sie einen Algorithmus, der bestimmt, ob G zusammenhängend ist, und begründen Sie seine Korrektheit.

Lösung zu Aufgabe 4.2
Gesucht ist der Breitensuche-Algorithmus:

- 1. Wähle einen beliebigen Knoten, hänge ihn hinten an Q an und markiere ihn. Initialisiere eine Zählvariable $k = 0$.
- 2. Nimm den ersten Knoten aus Q heraus und erhöhe k um 1. Hänge alle nicht-markierten Nachbarn dieses Knotens hinten an Q an und markiere sie. Falls Q jetzt nicht leer ist, wiederhole 2.
- Falls $k < |V|$, ist G nicht zusammenhängend, sonst ($k = |V|$) ist G zusammenhängend.

Alle Knoten beginnen unmarkiert. Wir betrachten die Zusammenhangskomponente von v. Jeder Knoten darin wird irgendwann während des Algorithmus genau einmal (als Nachbar eines markierten Knotens) entdeckt und markiert an Q angehängt. Wir beginnen mit v, und untersuchen durch die Funktionsweise von Q zuerst alle Knoten mit Abstand 1 zu v, dann alle mit Abstand 2 zu v, usw. Jeweils beim Herausnehmen des ersten Knoten aus Q wird k um 1 erhöht.Wir arbeiten bis die Warteschlange leer ist, entfernen also alle Knoten der

Zusammenhangskomponente von v, womit k genau die Anzahl der Knoten in der Zusammenhangskomponente von v repräsentiert.

Aufgabe 4.3
Sei G ein Graph, der durch die Knotenmenge V und eine Adjazenzliste N gegeben ist. Wie können Sie die Ideen der Algorithmen aus 4.1 und 4.2 so erweitern, dass G auf Kreisfreiheit getestet werden kann? Begründen Sie Ihre Antwort!

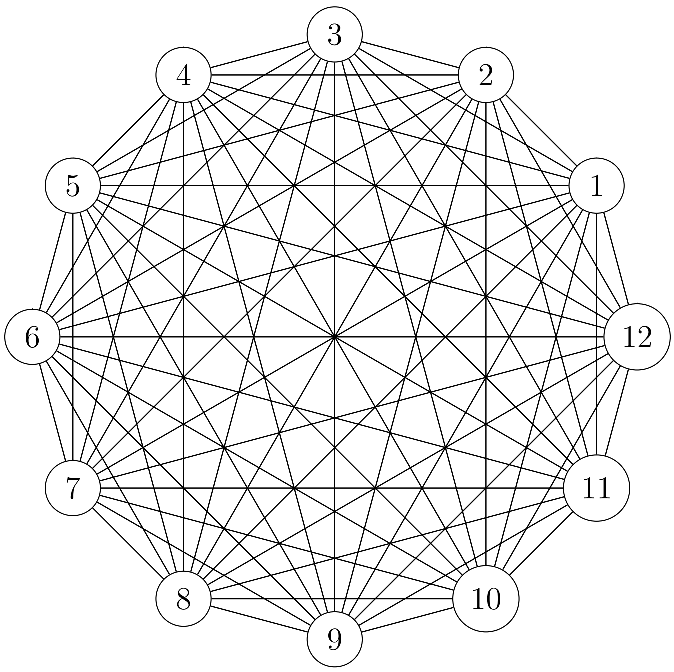
Lösung zu Aufgabe 4.3
Für jede Zusammenhangskomponente eines Graphen führt man den angegebenen Algorithmus 4.1 bzw. 4.2 durch, mit folgender Änderung:

Wir starten in jeder Zusammenhangskomponente mit einem zufälligen (bisher nicht-markierten) Knoten v. Beim Betrachten der Nachbarn jedes Knotens w (was in beiden Algorithmen genau einmal passiert), testet man dann, ob es darunter bereits zwei markierte Knoten $u' \neq u$ gibt. Ist das der Fall, dann gibt es eine Kante von w nach u und von w nach u' (jeweils Nachbar-Beziehung), und einen Weg von u nach u' in den markierten Knoten, da ein Weg von u nach v und ein Weg von u' nach v in den markierten Knoten existiert.

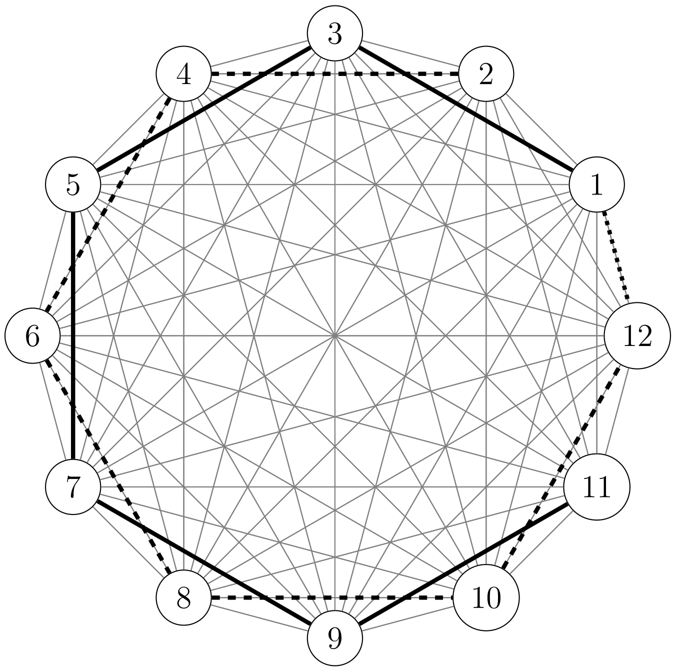
Aufgabe 4.4 *Hausaufgabe*
Gegeben sei der vollständige Graph $K_{12} = (V, E)$ auf $V = \{1, \ldots, 12\}$ mit Kantengewichten $w : E \to \mathbb{N}$ mit

$$w(\{i,j\}) := \begin{cases} 1, & \text{falls } i + j \equiv 0 \mod 2, \\ 2, & \text{falls } i + j \equiv 1 \mod 2. \end{cases}$$

Bestimmen Sie einen bezüglich w minimalen Spannbaum in diesem Graphen und beweisen Sie, dass Ihre Lösung tatsächlich minimal ist.



Lösung zu Aufgabe 4.4



Die Abbildung zeigt einen Spannbaum der Länge 12. Jeder Spannbaum enthält nach Vorlesung $12 - 1 = 11$ Kanten, hat also mindestens die Länge 11. Angenommen, es gäbe einen Spannbaum $T \subset E$ der Länge 11, dann enthielte dieser nur Kanten der Länge 1. Nach Definition der Kantengewichte w müssten für $\{u, v\} \in T$ also entweder u und v beide gerade oder beide ungerade Zahlen sein. Damit könnte es aber keinen Weg von (zum Beispiel) Knoten 1 nach Knoten 12 geben, da Knoten 1 nur mit ungeraden, Knoten 12 nur mit geraden Knoten verbunden wäre, und T wäre nicht zusammenhängend, Widerspruch. Der gezeigte Spannbaum ist also tatsächlich minimal.