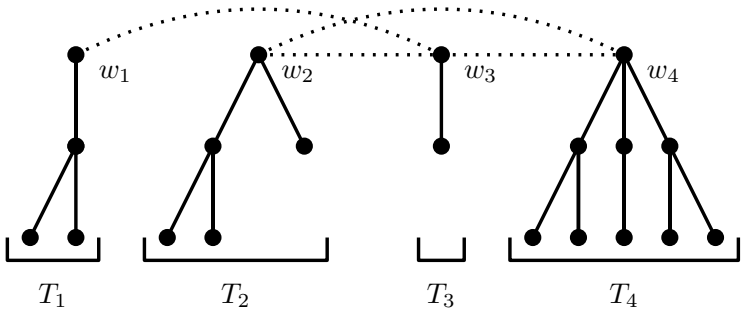
Aufgabe 1. Wir betrachten die Klasse $\mathcal{X}$ aller zusammenhängenden Graphen G, die durch folgende Konstruktion entstehen können. Sei $k \in \mathbb{N}$, seien $T_1 = (V_1, E_1), \ldots, T_k = (V_k, E_k)$ Bäume mit Wurzeln $w_1, \ldots, w_k$ und sei $E_w \subseteq \{w_1, \ldots, w_k\} \times \{w_1, \ldots, w_k\}$. Nehmen Sie an, dass $V_i \cap V_j = \emptyset$ für $i \neq j$ gilt. Dann ist G die Vereinigung über die Bäume, bei denen die Wurzeln wie von E_w beschrieben miteinander verbunden sind. Formal ist $G = (V, E)$ definiert durch:

$$V = \bigcup_{1 \leq i \leq k} V_i \quad \text{und} \quad E = E_w \cup \bigcup_{1 \leq i \leq k} E_i.$$

Hier ist ein Beispiel für $k = 4$. Die Kanten in E_w werden gepunktet dargestellt.



Erinnern Sie sich an das im Allgemeinen NP-schwere Problem 3-Coloring. Entwerfen und beschreiben Sie einen polynomiellen Algorithmus, der das Problem für den Spezialfall löst, dass G Teil dieser Graphklasse $\mathcal{X}$ ist. Nehmen Sie dabei an, dass $k \in \mathbb{N}$ **konstant** ist und dass die Knoten $w_1, \ldots, w_k$ explizit gegeben sind. Begründen Sie kurz die Korrektheit Ihres Algorithmus und erläutern Sie dessen Laufzeit.

- **Algorithmus**: Das Problem kann auf 3-Coloring für die Wurzelknoten reduziert werden, da alle anderen Knoten des Baum trivial anhand der Knotentiefe gefärbt werden können ($d\%2 = 1$: Andere Farbe wie Wurzel, $d\%2 = 0$: selbe Farbe wie Wurzel). Daher kann auch für den trivialen Fall $k \leq 3$ *true* zurückgegeben werden. Für $k > 3$ kann einfach das Ergebnis des NP-schweren 3-Coloring Algorithmus, ausgeführt auf die Wurzelknoten, zurückgegeben werden.
- **Korrektheit**: Da in einem Baum ein Knoten nur seinen Parent und seine Children als Nachbarn hat können Parent und Children die selbe Farbe haben und der Knoten selbst muss eine andere haben. Wenn also immer jede Ebene alternierend gefärbt wird ergibt sich eine valide Färbung.
- **Laufzeit**: Für die Färbung der Wurzelknoten ergibt sich eine Laufzeit von $O(2^k \cdot k)$. Da k eine Konstante ist ist $2^k \cdot k$ auch eine Konstante, womit sich eine Laufzeit von $O(1)$ ergibt. Wenn $G_w = (w_1, ..., w_k, E_w)$ gegeben ist, ist auch die Gesamtlaufzeit $O(1)$. Wenn wir dieses Subset erst konstruieren müssen ergibt sich aufgrund der Konstruktionsarbeit (machbar mit z.B. DFS) eine Laufzeit von $O(|V| + |E|)$.

Aufgabe 2. Wir definieren zwei Probleme: Für das Problem Cycle Pair besteht die Eingabe aus einem Graphen G und einer Zahl $\ell \in \mathbb{N}$ und es soll entschieden werden, ob es zwei knoten-disjunkte Kreise in G gibt, die jeweils Länge genau ℓ haben. Für das zweite Problem, nehmen Sie $k \in \mathbb{N}$ als konstant an. Dann ist k-Cycle das Problem für einen Graphen G zu entscheiden, ob G einen Kreis der Länge genau k enthält. Nehmen Sie an, dass P $\neq$ NP gilt.

(a) Zeigen Sie, dass Cycle Pair NP-schwer ist, indem Sie von einem verwandten, aus der Vorlesung bekannten Problem reduzieren.

(b) Entwerfen und beschreiben Sie einen polynomiellen Algorithmus für k-Cycle. Begründen Sie kurz dessen Korrektheit und polynomielle Laufzeit.

(c) Begründen Sie, ob 3-Color $\leq_P$ Cycle Pair gilt.

(d) Begründen Sie, ob 3-Color $\leq_P$ k-Cycle gilt.

Hinweis: Für die Teilaufgaben (c) und (d) müssen Sie keine Reduktionen angeben.

(a) Ham-Cycle $\leq_P$ Cycle Pair:
Man dupliziert den Graphen $G = (V, E)$ und verbindet genau einen Knoten der beiden Graphen. Der resultierende Graph G' enthält genau dann ein Cycle Pair der Länge $|V|$ wenn der ursprüngliche Graph einen Hamiltonkreis enthält.

(b) Man führt den DFS-Algorithmus aus und nummeriert die besuchten Knoten aufsteigend durch. Sollte man an einem Punkt einen bereits besuchten Knoten nochmal sehen ergibt sich daraus, dass der Graph einen Kreis der Länge $|n_2 - n_1| + 1$ ergibt, wobei n_2 und n_1 die zugewiesenen Nummern der Knoten sind. Wenn ein Kreis der Länge k gefunden wird geben wir *true* zurück, wenn der ganze Graph durchsucht wurde $false$.
Polynomiell weil DFS in $O(|V| + |E|)$ ausführbar ist.

(c) Da Ham-Cycle NP-complete ist können alle Probleme in NP (inklusive 3-color) auf Ham-Cycle reduziert werden, also gilt 3-color $\leq_P$ Ham-Cycle. Da Reduktionen transitiv sind und Ham-Cycle $\leq_P$ Cycle Pair gilt, gilt auch 3-color $\leq_P$ Cycle Pair.

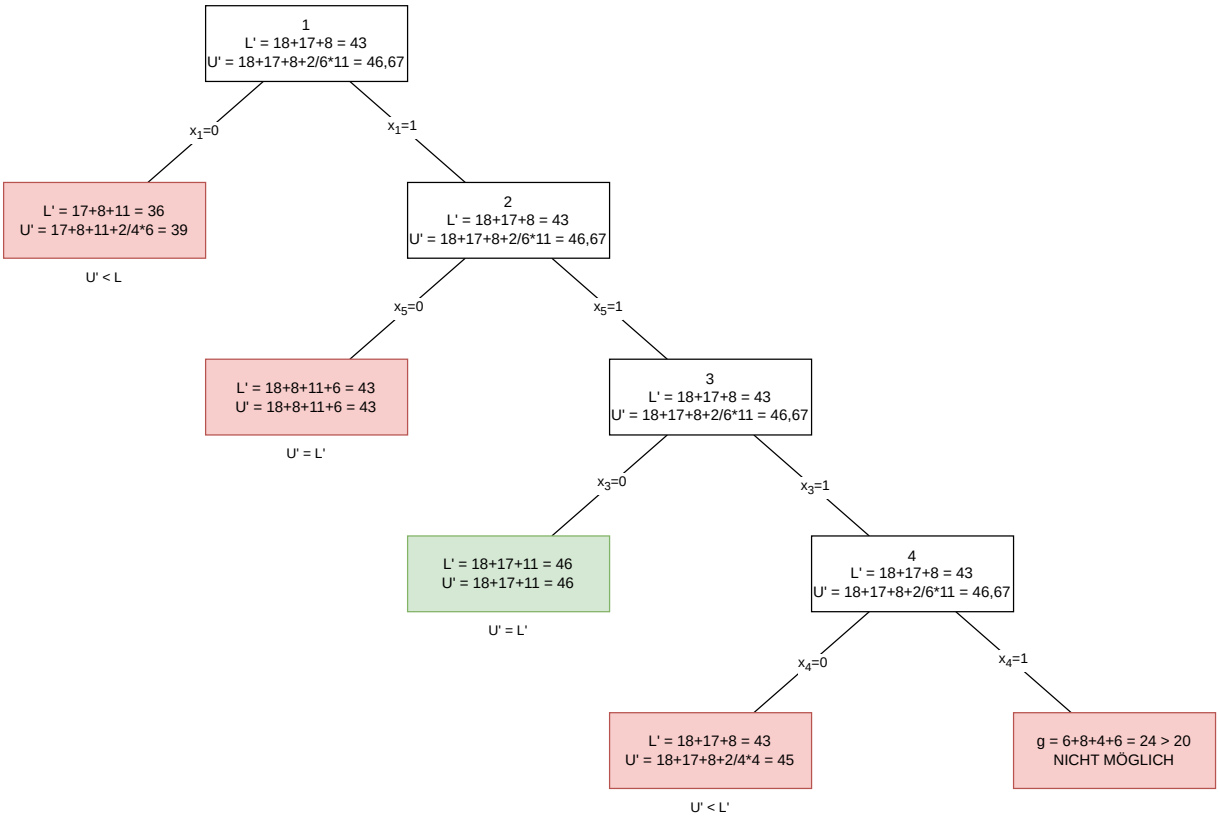
(d) Da $P \neq NP$ und k-Cycle in P liegt und 3-color NP-complete ist, kann 3-color nicht auf k-Cycle reduziert werden, da ansonsten alle Problemem in NP auf k-Cycle reduziert werden können und $P = NP$ gelten würde.

Aufgabe 3. Wenden Sie den Branch-and-Bound Algorithmus aus der Vorlesung auf die unten angegebene Instanz des Rucksackproblems an. Stellen Sie den Ablauf des Algorithmus als Baum dar. Geben Sie für jeden Schritt die obere Schranke U', die untere Schranke L', sowie eine passende Auswahl von Gegenständen mit Gesamtwert L' an. Verwenden Sie die „Best-First“ Heuristik für die Auswahl von Teilproblemen.

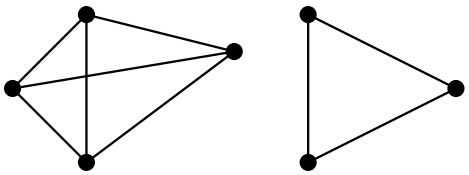
Gegenstand	1	2	3	4	5
Gewicht g_i	6	4	4	6	8
Wert w_i	18	6	8	11	17

Rucksackkapazität: 20

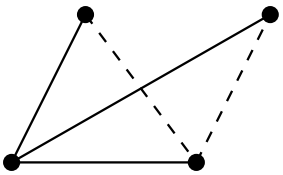
Reihenfolge (nach $\frac{w_i}{g_i}$): 1 ($\frac{18}{6} = 3$), 5 ($\frac{17}{8} = 2.125$), 3 ($\frac{8}{4} = 2$), 4 ($\frac{11}{6} = 1.8\dot{3}$), 2 ($\frac{6}{4} = 1.5$).



Aufgabe 4. Entwerfen und beschreiben Sie einen Branch-and-Bound Algorithmus für das Optimierungsproblem Cluster Editing, welches wie unten definiert ist. Dabei ist ein *Clustergraph* ein Graph, bei dem jede Zusammenhangskomponente eine Clique ist. Man kann leicht zeigen, dass ein Graph G ein Clustergraph ist, genau dann wenn G keinen Pfad mit drei Knoten als induzierten Teilgraphen enthält. Hier ist ein Beispiel:



Clustergraph



Kein Clustergraph mit einem gestrichelt hervorgehobenem induzierten Pfad mit drei Knoten.

Cluster Editing

Eingabe: Ein (schlichter) Graph G.

Aufgabe: Kanten zu G hinzufügen oder aus G entfernen sodass der resultierende Graph ein Clustergraph ist.

Optimierungsziel: Die Anzahl der hinzugefügten und entfernten Kanten minimieren.

Gehen Sie in Ihrer Lösung insbesondere auf nachfolgende Punkte ein. Pseudocode ist nicht notwendig.

(a) Wie repräsentieren Sie ein Teilproblem?

(b) Wie erfolgt das Branching, d.h. das Aufteilen eines (Teil-)Problems in weitere Teilprobleme?

(c) Beschreiben Sie eine sinnvolle Möglichkeit für die Auswahl des nächsten Teilproblems. Nach welcher Strategie gehen Sie dabei vor?

In dieser Aufgabe können Sie sich auf die Teilprobleme und das Branching konzentrieren. Es ist nicht nötig auf obere und untere Schranken einzugehen.

(a) Hat der aktuelle Knoten mit zwei beliebigen anderen Knoten einen induzierten Teilgraph der ein Pfad ist?

(b) Durch Löschen einer der beiden Kanten des Pfads oder hinzufügen einer Kante, dass der Pfad zu einem Kreis wird.

(c) Als nächstes Teilproblem sollte immer das Teilproblem mit der niedrigsten unteren Schranke gewählt werden (Best first).