Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Peter Sanders
L. Hübschle-Schneider, T. Maier

10. Übungsblatt zu Theoretische Grundlagen der Informatik im WS 2015/16

http://algo2.iti.kit.edu/TGI2015.php
{sanders,huebschle,t.maier}@kit.edu

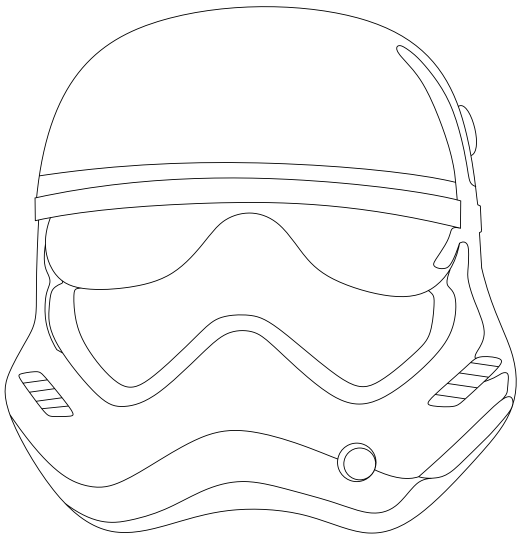
Musterlösungen

Keine Panik, das letzte abzugebende Blatt war Blatt 8 — Blatt 9 war das Weihnachtsblatt!

Aufgabe 1 (*Instanzen NP-vollständiger Probleme,* $1 + 2 + 2$ *Punkte*)

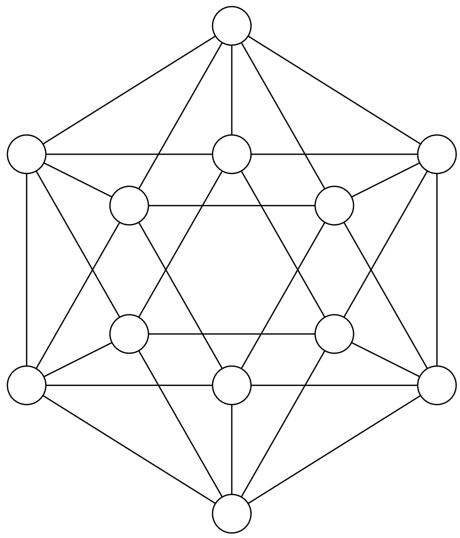
a) 4-COLOR: gegeben einem Graph, lassen sich die Knoten des Graphen mit 4 Farben so färben, dass keine 2 benachbarten Knoten die selbe Farbe tragen.

Färben sie den gegebenen Flächengraph, mit 4 Farben (keine zwei benachbarten Flächen dürfen die selbe Farbe tragen).



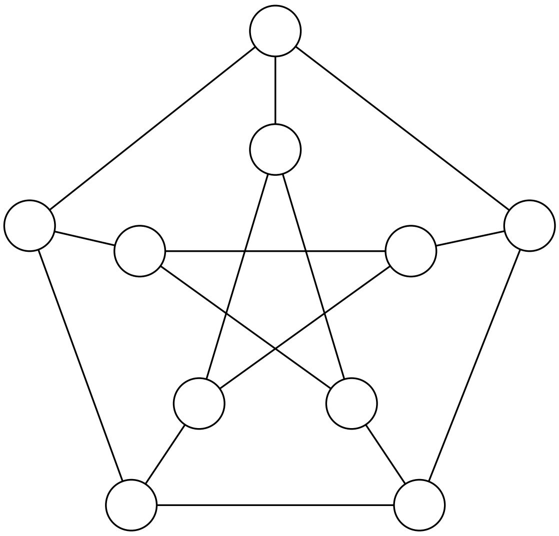
b) HAMILTONIAN CYCLE: gegeben einem Graph, existiert ein einfacher Zyklus, der jeden Knoten genau einmal besucht (streng genommen wird der Startknoten zwei mal besucht).

Zeichnen sie in dem folgenden Graph zwei Kantendisjunkte Hamilton-Kreise ein.



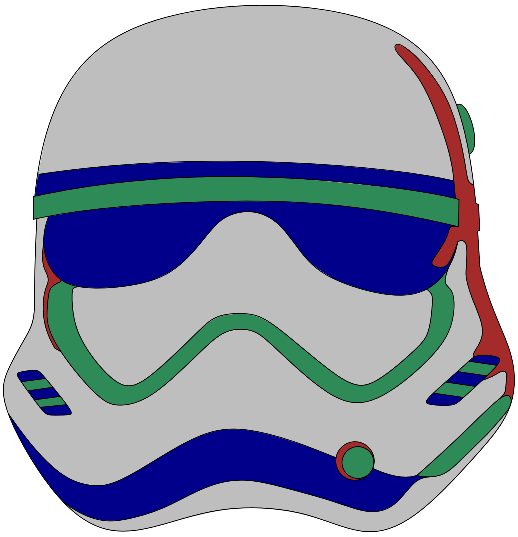
c) VERTEX COVER: gegeben einem Graph $G = (V, E)$ und einer Zahl k, existiert eine Knotenmenge $C \subseteq V$ mit $|C| \leq k$, so dass jede Kante des Graphen, zu mindestens einem Knoten aus C inzident ist.

Zeichnen sie in dem gegebenen Graphen ein minimales VertexCover ein.

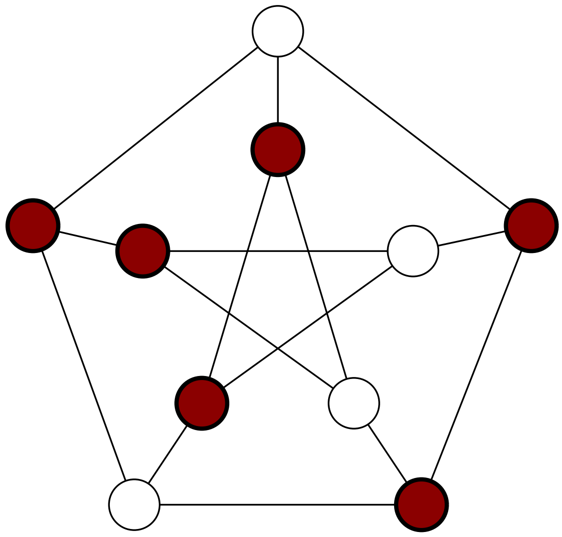


Musterlösung:

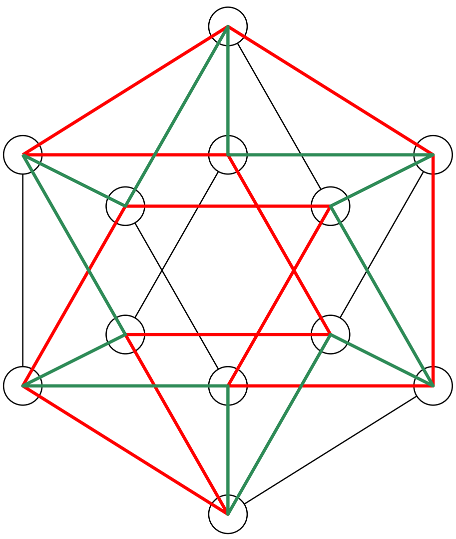
a)



c)



b)



Aufgabe 2 (*Reduktionen,* $3 + 3$ *Punkte*)

a) Geben Sie eine *lineare* Reduktion von kSUCCESSORS auf SORT an. Dabei ist SORT das Sortierproblem über den natürlichen Zahlen. Das Entscheidungsproblem kSUCCESSORS fragt ob gegeben einer Menge $M \subset \mathbb{N}$ und einer Zahl $k \in \mathbb{N}$ eine Folge von (mindestens) k aufeinanderfolgenden Zahlen in M existiert. Beispielsweise ist das Problem für $M = \{4, 1, 3, 9, 8, 5\}$ und $k = 3$ lösbar, da $\{3, 4, 5\} \subseteq M$.

b) Geben Sie eine *lineare* Reduktion von SUBSET SUM auf KNAPSACK an. Das Problem SUBSET SUM ist folgendermaßen definiert: Gegeben eine Menge von n Gegenständen, habe Gegenstand i das Gewicht $w_i \in \mathbb{N}$. Gegeben sei weiterhin ein Parameter $W \in \mathbb{N}$. Gibt es eine Teilmenge $M \subseteq G$ sodass $\sum_{i \in M} w_i = W$?

Musterlösung:

a) Als erstes sortieren wir M. Anschließend suchen wir mit einem linearen Scan Folgen aufeinanderfolgender Zahlen. Dazu merken wir uns die aktuelle und bisher längste gefundene solche Folge:

Function kSUCC($k \in \mathbb{N}, M \subset \mathbb{N}$)
    sort(M);   −− Benutze Algorithmus für das Problem SORT
    curr $= 0$, longest $= 0 : \mathbb{N}$;
    for $i := 1$ **to** $|M| - 1$:   −− $\mathcal{O}(1)$ pro Element, ein Durchlauf pro Element
        if $M[i] = M[i-1] + 1$:
            curr $=$ curr $+ 1$;
        else
            longest $=$ max(longest, curr);
            curr $= 0$;
    return longest $\geq k$;

Der gegebene Algorithmus kSUCC ist also eine lineare Reduktion von kSUCCESSORS auf SORT.

b) Wir konstruieren aus der Instanz von SUBSET SUM wie folgt eine KNAPSACK-Instanz: Die Gegenstände der SUBSET SUM-Instanz übernehmen wir und weisen ihnen genau ihr Gewicht als Profit zu: $p_i = w_i$. Dann lösen wir die KNAPSACK-Instanz mit Gewichtslimit W und Zielprofit W. Die Instanz von SUBSET SUM ist genau dann lösbar, wenn die KNAPSACK-Instanz lösbar ist:

- Wenn die SUBSET SUM-Instanz lösbar ist, dann ist auch die von der Reduktion erzeugte KNAPSACK-Instanz lösbar, denn es gibt eine Menge M von Gegenständen mit Gesamtgewicht W, die die SUBSET SUM-Instanz löst. Da der Profit jedes Gegenstandes gleich seinem Gewicht ist, hat M Gesamtprofit W und löst daher die KNAPSACK-Instanz.
- Wenn die von der Reduktion erzeugte KNAPSACK-Instanz lösbar ist, dann ist auch die ursprüngliche SUBSET SUM-Instanz lösbar. Sei M die Menge der in den Rucksack gepackten Gegenstände, deren Gesamtgewicht höchstens W und Profit mindestens W ist. Anders formuliert: $\sum_{i \in M} w_i \leq W$ und $\sum_{i \in M} w_i \geq W$, also gleich W. Daher löst M die SUBSET SUM-Instanz.

Aufgabe 3 (NP, $1 + 1 + 1 + 1$ *Punkte*)

Liegen die folgenden Probleme in NP? Beweisen Sie Ihre Antwort!

a) SORT, das Sortieren einer Menge

b) TSP, das Entscheidungsproblem des Handlungsreisenden.

c) SHORTESTPATH, das Problem des kürzesten Weges. Genauer: Gegeben seien ein Graph $G = (V, E)$, zwei Knoten $s, t \in V$ und eine Maximallänge ℓ. Existiert ein Pfad $p = \langle s, v_1, \ldots, v_{n-1}, t \rangle$ in G, sodass die Summe der Kantengewichte auf dem Pfad ℓ nicht überschreitet?

d) CFG REGULARITY, das Regularitätsproblem für kontextfreie Grammatiken. Gegeben sei eine kontextfreie Grammatik G über dem Alphabet Σ, ist $L(G)$ eine reguläre Sprache?

Musterlösung:

a) Das Problem ist kein Entscheidungsproblem, kann also nicht in NP liegen, denn NP ist eine Klasse von Entscheidungsproblemen.

b) Ja, denn wir können einfach überprüfen ob der Pfad alle Knoten besucht und ob seine Länge die gegebene Maximallänge nicht überschreitet. Wichtig ist, dass es sich hier um die Formulierung als Entscheidungsproblem handelt! Festzustellen, ob der Pfad der *kürzeste* Pfad ist, der alle Knoten besucht, ist nämlich weitaus schwieriger: Bisher ist kein Polynomialzeitalgorithmus dafür bekannt.

c) Ja, denn mit Dijkstra's Algorithmus lässt sich leicht in Polynomialzeit verifizieren, ob ein gegebener s-t-Pfad der kürzeste Pfad von s nach t ist.

d) Nein, denn das Problem ist unentscheidbar (siehe Kapitel 1.3 der Vorlesung).