Übungsblatt 3

Abgabe bis 4. Dezember 2024

Organisatorisches:

- Für das Lösen der Übungsaufgaben habt ihr *zwei Wochen* Zeit.
- Die Bearbeitung soll möglichst in Zweiergruppen erfolgen.
- Die Abgabe der Lösungen als *schönes* PDF erfolgt per Email an `paramalgo_abgabe@lists.kit.edu`
- **Bonuspunkte:** Die Bonuspunkte sind übertragbar auf andere Übungsblätter.

Aufgabe 1: Odd Subgraph — 3 + 5 Punkte

Gegeben ist ein Graph G und ein Parameter k. Das Problem Odd Subgraph fragt, ob es einen Subgraphen von G mit k Kanten gibt, in dem alle Knoten ungeraden Grad haben.

Teilaufgabe (a) Zeige: Wenn der maximale Knotengrad in G höchstens k ist und G mindestens $k \cdot (2k-1)$ Kanten enthält, dann gibt es ein Matching in G mit k Kanten.

Teilaufgabe (b) Zeige, dass für Odd Subgraph ein Kern der Größe $O(k^2)$ in polynomieller Zeit berechenbar ist.
Tipp: Wie löst du die Fälle, die von Teilaufgabe (a) nicht abgedeckt sind?

Aufgabe 2: Edge Clique Cover — 10 Punkte

Das parametrisierte Problem Edge Clique Cover ist wie folgt definiert. Gegeben ist ein Graph G sowie ein Parameter k. Gesucht ist eine Menge von maximal k Cliquen, sodass jede Kante von G in mindestens einer der Cliquen enthalten ist.

Gib sichere Reduktionsregeln an, die für Edge Clique Cover einen Kern mit maximal 2^k Knoten berechnen.

Hinweis: Zwei Knoten u und v sind *echte Zwillinge*, wenn $N(u) \cup \{u\} = N(v) \cup \{v\}$ (dabei bezeichnet $N(u)$ die Menge aller Nachbarn von u). Kannst du solche echten Zwillinge loswerden?

Aufgabe 3: Anwendung von Lenstra — 4 + 8 Punkte

Erstelle eine geeignete ILP Formulierung, um zu zeigen, dass folgende Probleme in FPT sind:

Variety Subset Sum: Gegeben eine Multimenge A von Zahlen in $\mathbb{N}$ ("Multi" heißt, Zahlen können mehrfach in A auftauchen) und ein Wert $b \in \mathbb{N}$. Gibt es eine Teilmultimenge $X \subseteq A$ mit $\sum_{x \in X} x = b$? Betrachte als Parameter die Anzahl unterschiedlicher Zahlen in A.

Makespan Scheduling: Gegeben $m \in \mathbb{N}$ Maschinen, n Jobs mit Bearbeitunszeiten $p_1, \dots, p_n \in \mathbb{N}$ und eine Schranke für die maximale Bearbeitungsdauer $k \in \mathbb{N}$. Gibt es eine Verteilung der Jobs auf die Maschinen, sodass keine Maschine länger als k Zeit benötigt? (Ganz formal: Gibt es eine Abbildung $f : \{1, \dots, n\} \to \{1, \dots, m\}$ sodass $\sum_{j : f(j)=i} p_j \leq k$ für alle $1 \leq i \leq m$?)

Betrachte als Parameter die Schranke k auf die Bearbeitungsdauer.

Hinweis: Makespan Scheduling benötigt eine ähnliche Strategie wie das ILP für Closest String aus der Übung (am 28.11). Folgende Überlegung könnte nützlich sein: Welche Kosten können die Jobs haben und wie lassen sich die möglichen Jobzuweisungen auf eine Maschine mit einer Funktion in k beschränken?

Aufgabe 4: Closest String — 5 Bonus-Punkte

In dieser Aufgabe sollst du in einer Programmiersprache deiner Wahl ein Programm implementieren, das das Problem Closest String löst. Gegeben ist eine Menge von Strings, gesucht ist ein minimales k und ein String s, sodass s zu jedem gegebenen String Hamming-Distanz höchstens k hat.

Nutze dazu den Algorithmus aus der Aktiv-Session 1. Du darfst aber gerne auch weitere Regeln implementieren und Optimierungen machen.

Beschreibe in der PDF-Abgabe, welche Verfahren du implementiert hast und was du noch weiter optimiert hast. Gib außerdem in der PDF-Abgabe für jede Instanz das kleinste k an, für das du eine Lösung gefunden hast.

Gib zusätzlich den Quellcode sowie deine gefundenen Lösungen (im unten beschriebenen Format) als eine ZIP-Datei ab.

Für jede gelöste Instanz kannst du einen Punkt bekommen.

Dateiformat Eingabe: In der ersten Zeile steht n, die Anzahl der Strings. In den nächsten n Zeilen ist jeweils ein String gegeben. Die Strings haben alle die gleiche Länge.

Dateiformat Ausgabe: Eine Zeile mit einem String, der minimale Hamming-Distanz zu den gegebenen Strings hat.

Hinweis: Mithilfe der Datei `validator.py` kannst du die Hamming-Distanz deiner Lösung zu den gegebenen Strings bestimmen. Dabei wird das Maximum über alle Hamming-Distanzen ausgegeben.

Aufgabe 5: Kontaktvermeidung **8 nicht übertragbare Bonuspunkte**

Ein Virus geht um und muss dringend an der Ausbreitung gehindert werden. Laut Modellrechnungen kann dies bewerkstelligt werden, indem sich jeder nur noch mit (höchstens) d anderen Leuten trifft. Zum Glück können bis zu k Personen geimpft werden, was dazu führt, dass deren Kontakte sofort ignoriert werden können.

Es muss nun also entschieden werden, ob es möglich ist k Knoten aus einem gegebenen Graphen zu löschen, so dass anschließend der Maximalgrad höchstens d ist.

Gib sichere Reduktionsregeln an, die für dieses Problem einen Kern mit Größe polynomiell in $d + k$ berechnen.