TU Wien
Institut für Logic and Computation
Forschungsbereich Algorithms and Complexity

186.866 Algorithmen und Datenstrukturen VU

Übungsblatt 7

PDF erstellt am: 30. Mai 2023

Deadline für dieses Übungsblatt ist **Montag, 19.06.2023, 20:00 Uhr**. Um Aufgaben für diese Übung anerkannt zu bekommen, gehen Sie folgendermaßen vor:

1. Öffnen Sie den TUWEL-Kurs der Lehrveranstaltung *186.866 Algorithmen und Datenstrukturen (VU 5.5)* und navigieren Sie zum Abschnitt *Übungsblätter*.
2. Teilen Sie uns mit, welche Aufgaben Sie gelöst haben **und** welche gelösten Aufgaben Sie gegebenenfalls in der Übungseinheit präsentieren können. Gehen Sie dabei folgendermaßen vor:
   - Laden Sie Ihre Lösungen in einem einzigen PDF-Dokument in TUWEL hoch.
     Link *Hochladen Lösungen Übungsblatt 7*
     Button *Abgabe hinzufügen*
     PDF-Datei mit Lösungen hochladen und *Änderungen sichern*.
   - Kreuzen Sie an, welche Aufgaben Sie gegebenenfalls in der Übung präsentieren können. Die Lösungen der angekreuzten Aufgaben müssen im hochgeladenen PDF enthalten sein.
     Link *Ankreuzen Übungsblatt 7*
     Button *Abgabe bearbeiten*
     Bearbeitete Aufgaben anhaken und *Änderungen speichern*.

Bitte beachten Sie:

- Bis zur Deadline können Sie sowohl Ihr hochgeladenes PDF, als auch Ihre angekreuzten Aufgaben beliebig oft verändern. Nach der Deadline ist keine Veränderung mehr möglich. Es werden ausnahmslos keine Nachabgabeversuche (z.B. per E-Mail) akzeptiert.
- Sie können Ihre Lösungen entweder direkt in einem Textverarbeitungsprogramm erstellen, oder aber auch gut leserliche Scans bzw. Fotos von handschriftlichen Ausarbeitungen hochladen (beachten Sie die maximale Dateigröße).
- Beachten Sie die Richtlinien für das An- und Aberkennen von Aufgaben (Details finden Sie in den Folien der Vorbesprechung).

Aufgabe 1. Gegeben ist folgende Instanz des Rucksackproblems:

Kapazität $G = 10$

#	Wert	Gewicht
1	1	2
2	9	3
3	14	3
4	19	4
5	22	5
6	40	8

(a) Wenden Sie den aus der Vorlesung bekannten Greedy-Algorithmus für das Rucksackproblem auf die obige Instanz an. Welche Gegenstände werden ausgewählt? Welchen Gesamtwert haben diese?

(b) Lösen Sie die obige Instanz jetzt durch dynamische Programmierung. Geben Sie dazu, wie aus der Vorlesung bekannt, die vollständige Belegung der 2-dimensionalen Lösungstabelle an.

Geben Sie die Menge der für die optimale Lösung ausgewählten Gegenstände an und markieren Sie in der Tabelle durch Einkreisen all jene Felder, die der Algorithmus `Find-Solution` aus der Vorlesung bei der Berechnung der Lösungsmenge A ausliest und verwendet. Welchen Gesamtwert hat die optimale Lösung?

	0	1	2	3	4	5	6	7	8	9	10
$\emptyset$											
$\{1\}$											
$\{1, 2\}$											
$\{1, 2, 3\}$											
$\{1, 2, 3, 4\}$											
$\{1, 2, 3, 4, 5\}$											
$\{1, 2, 3, 4, 5, 6\}$											

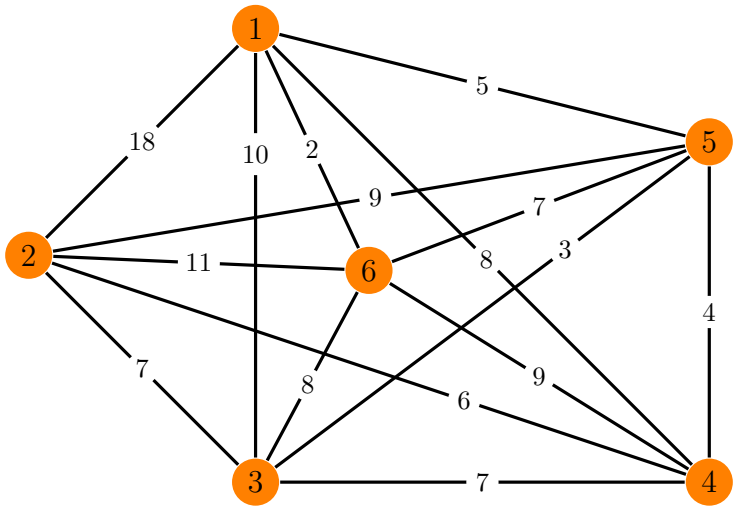
Aufgabe 2. Eine DNA-Sequenz ist eine Folge der vier Basen Adenin (A), Cytosin (C), Guanin (G) und Thymin (T). Ein Beispiel für eine DNA-Sequenz der Länge 9 ist etwa $ATCGATTAC$.

Gegeben sind zwei DNA-Sequenzen $X = x_1 \dots x_m$ und $Y = y_1 \dots y_n$. Entwerfen Sie einen Algorithmus, der mittels dynamischer Programmierung die Länge der längsten DNA-Sequenz ermittelt welche jeweils eine **zusammenhängende** Teilfolge von X und von Y ist. Die Laufzeit des Algorithmus soll in $O(mn)$ liegen.

Beispiel: Bei Eingabe $X = ATCGAT$ und $Y = AGATGC$ ist GAT die längste gemeinsame zusammenhängende Teilfolge und es soll 3 zurückgegeben werden.

Aufgabe 3. Gegeben ist eine Variante des Rucksackproblems, bei der jeder Gegenstand beliebig oft gewählt werden kann. Zeigen Sie, dass der Greedy-Algorithmus aus der Vorlesung für dieses Problem eine Gütegarantie von $\frac{1}{2}$ erzielt. Nehmen Sie an, dass das Gewicht eines jeden Gegenstands die Rucksack-Kapazität nicht überschreitet.

Aufgabe 4. Gegeben sei eine Instanz des symmetrischen Traveling Salesperson Problems (TSP) in Form des folgenden vollständigen Graphen G mit Knoten $V = \{1, 2, 3, 4, 5, 6\}$ und Kanten E, denen die eingezeichneten Gewichte zugeordnet sind:



(a) Wenden Sie auf diese Instanz *zweimal* die in der Vorlesung gezeigte Spanning-Tree-Heuristik an, wobei Sie unterschiedliche Eulerkreise verwenden, sodass zwei unterschiedliche Touren erzeugt werden. Beschreiben Sie die einzelnen Schritte im Detail.

(b) Versuchen Sie eine optimale Tour durch „Hinsehen“ und Ihre Intuition zu finden. Wieviele Touren müssten Sie mindestens allgemein für das symmetrische TSP mit n Städten und hier konkret in einem vollständigen Enumerations-Verfahren durchprobieren, um eine bewiesen optimalen Lösungen zu finden?

Hinweis: $n!$ kann durch Ausnutzung von Symmetrien verbessert werden.

(c) Ist die gegebene Instanz metrisch? Was bedeutet das im Allgemeinen für die Gütegarantie der Spanning-Tree-Heuristik? Welche konkreten Approximationsgüten haben Ihre beiden in Unteraufgabe (a) gefundenen Touren?

(d) **Optional, für Interessierte:** Skizzieren Sie einen Algorithmus der die Länge der kürzesten Tour in einem beliebigen vollständigen Graphen mit n Knoten und Distanzmatrix c in Zeit $O(2^n n^4)$ findet.

Hinweis 1: Betrachten Sie alle Teilmengen von Knoten.

Hinweis 2: Dynamische Programmierung.

Aufgabe 5. Für einen gerichteten Graphen $G = (V, E)$ wird eine größtmögliche (bezüglich Kardinalität) Teilmenge $E' \subseteq E$ von Kanten gesucht, sodass der Teilgraph $G' = (V, E')$ azyklisch ist. Geben Sie für dieses Problem einen polynomiellen Approximationsalgorithmus mit Gütegarantie $\frac{1}{2}$ an und beweisen Sie, dass Ihr Algorithmus diese Gütegarantie einhält.

Hinweis: Betrachten Sie eine beliebige lineare Ordnung der Knotenmenge. Welche zwei Arten von Kanten können Sie unterscheiden?

Aufgabe 6. Im MAX-k-PART-Problem wird für einen Graphen $G = (V, E)$ nach einer Partitionierung der Knoten V in k paarweise disjunkte Mengen $V_1, V_2, \ldots, V_k$ mit $\bigcup_{i=1}^{k} V_i = V$ gesucht, sodass die Anzahl der sogenannten *Verbindungskanten* maximiert wird. Eine Kante $(a, b) \in E$ ist dabei eine Verbindungskante, wenn $i, j \in \{1, \ldots, k\}$ mit $i \neq j$ existieren, sodass $a \in V_i$ und $b \in V_j$.

Sei $v_1, \ldots, v_n$ eine beliebige Ordnung der Knoten von G. Betrachten Sie den folgenden Greedy-Algorithmus. Zunächst weisen wir v_1 der Menge V_1 zu. Jeden weiteren Knoten v_i, $2 \leq i \leq n$, weisen wir der jeweiligen Knotenmenge zu, für die die Anzahl neuer Verbindungskanten am größten ist (bei Gleichheit ist die Zuweisung beliebig).

Zeigen Sie, dass dieser Algorithmus eine Partitionierung der Knoten findet, die mindestens $1 - \frac{1}{k}$ so viele Verbindungskanten hat wie eine optimale Partitionierung.

Hinweis: Weisen Sie jeder Kante den Endknoten mit größerem Index als *verantwortlichen* Knoten zu; dieser Knoten entscheidet, ob die Kante eine Verbindungskante ist. Sei r_i die Anzahl der Kanten, für die v_i verantwortlich ist. Argumentieren Sie, dass bei der Zuweisung von v_i im Algorithmus mindestens $r_i - \frac{r_i}{k}$ Verbindungskanten hinzukommen, und vervollständigen Sie das Argument.
