Ausgabedatum: 17.01.2013
Version: 2013-01-17 11:47

13. Übungsblatt zur Vorlesung
Entwurf und Analyse von Algorithmen, WS 12/13

Abgabe: *Bis Donnerstag, 24.01.2013, 12:00 Uhr, Abgabenkasten im Treppenhaus 48-6.*

Wichtig: Bitte denken Sie an die Vorlesungsumfrage!

Damit wir die Vorlesung weiterentwickeln können, brauchen wir Ihr Feedback! Dabei ist wichtig, dass sowohl *konstruktive* Kritik geübt als auch das Gute genannt wird, damit wir einen repräsentativen Eindruck Ihrer Meinungen erhalten. Leider haben bisher nur etwa 20% von Ihnen an der VLU teilgenommen.

Teilen Sie uns also bitte in jedem Fall Ihre Meinung unter

`http://vlu.informatik.uni-kl.de`

mit. Zugangstokens gingen Ihnen per Mail zu; wenn nicht, wenden Sie sich bitte an das SCI.

Danke!

83. Aufgabe

Track ε [4]
Track AI [4]

Gegeben ist eine Menge von n Punkten in der Ebene. Entwerfen Sie einen Algorithmus mit einer Laufzeit in $o(n^2)$ (kein Tippfehler, kleines o), der ein Paar von Punkten mit kleinster euklidischer Distanz unter allen Paaren bestimmt. Beweisen Sie, dass Ihr Algorithmus die geforderte Laufzeitschranke erfüllt.

84. Aufgabe

	a)	b)	c)
Track ε	1	2	[2]
Track AI	1	2	[2]

Wir betrachten binäre Suchbäume und gehen davon aus, dass uns die Schlüssel $a_1 < a_2 < \cdots < a_n$ zusammen mit ihren Zugriffswahrscheinlichkeiten $p_1, p_2, \ldots, p_n$ gegeben sind. Gesucht ist eine Suchbaumstruktur, die die erwartete Suchzeit für die Schlüssel a_i, also die Summe $\sum_{1 \leq i \leq n} p_i t_i$ minimiert, wobei t_i das Niveau des Schlüssel a_i im konstruierten Baum darstellt.

a) Begründen Sie, warum das BELLMANsche Optimalitätskriterium für dieses Problem erfüllt ist, sich die dynamische Programmierung also zur Optimierung eignet.

b) Entwerfen Sie mittels der dynamischen Programmierung einen Algorithmus, der optimale Binäre Suchbäume konstruiert.

   Begründen Sie die Korrektheit und bestimmen Sie die asymptotische Worst-Case Laufzeit (als Θ-Klasse) Ihres Algorithmus.

c) Leiten Sie eine obere Schranke für die mittlere Tiefe (gewichtet mit den Wahrscheinlichkeiten p_i) der Schlüssel in einem optimalen Suchbaum her.

   Hinweis: Die Entropie ist dafür ein geeignetes Hilfsmittel.

85. Aufgabe

	a)	b)
Track ε	[2]	[3]
Track AI	2	[3]

Sei $x = x_1, \ldots, x_n$ eine Folge von natürlichen Zahlen. Wir nennen

$$\mathrm{x}_{i_1}, \ldots, x_{i_k}$$

mit $i_j \in [1..n]$ für alle $j \in [1..k]$ eine *monotone Teilfolge* von x, wenn für alle $j \in [1 : k-1]$ gilt, dass

$$i_j < i_{j+1} \quad \text{und} \quad x_{i_j} < x_{i_{j+1}} \ .$$

a) Geben Sie einen Algorithmus an, der mit Hilfe dynamischer Programmierung in Zeit $\mathcal{O}(n^2)$, gegeben eine beliebige solche Folge x, eine längste monotone Teilfolge von x bestimmt.

   Begründen Sie die Korrektheit Ihres Algorithmus und dass er die Laufzeitschranke einhält.

b) Geben Sie einen Algorithmus an, der in Zeit $\mathcal{O}(n \log n)$, gegeben eine beliebige solche Folge x, eine längste monotone Teilfolge von x bestimmt.

   Begründen Sie die Korrektheit Ihres Algorithmus und dass er die Laufzeitschranke einhält.

   Hinweis: Betrachten Sie die monotonen Teilfolgen der Länge k mit dem jeweils kleinstmöglichen letzten Element.

86. Aufgabe

	a)	b)
Track ε	2	[2]
Track AI	[2]	[2]

a) Sei S eine endliche Menge mit $|S| \geq 2$. Sei $S_1, \dots, S_k$ eine Partition von S. Sei weiterhin $\mathcal{U} = \{A \subseteq S \mid |A \cap S_i| \leq 1, 1 \leq i \leq k\}$.

Zeigen Sie: $(S, \mathcal{U})$ ist ein Matroid.

b) Sei V ein Vektorraum, $M \subset V$ eine endliche Teilmenge dessen und

$$\mathrm{Ind}_M = \{A \subseteq M \mid A \text{ linear unabhängig } \}$$

die Menge aller linear unabhängigen Mengen von Vektoren aus M.

Zeigen Sie: (M, Ind_M) ist (stets) ein Matroid.

87. Aufgabe

	a)	b)	c)
Track ε	3	[2]	[1]
Track AI	3	[2]	[1]

Wir konstruieren einen Geld-Automaten, der bei Abbuchung des Betrages n (in Eurocent) diesen in einer möglichst kleine Anzahl Münzen auszahlt.

a) Geben Sie einen Greedy-Algorithmus an, der die optimale Stückelung (Anzahl 1 Cent-Münzen, Anzahl 2 Cent-Münzen, ...) auf Basis der für den Euro zur Verfügung stehenden Münzen berechnet.

Begründen Sie die Korrektheit des Algorithmus.

b) Nehmen Sie an, die Wertigkeit der Münzen sei durch b^i für eine ganze Zahl $b > 1$ und $0 \leq i \leq k$, $k \geq 1$ fest, gegeben. Beweisen Sie, dass ein Greedy-Algorithmus stets eine optimale Stückelung berechnen kann.

c) Gibt es Münzsätze (Menge verschiedener Wertigkeiten), für die Greedy-Algorithmen an der Berechnung einer optimalen Stückelung scheitern können? Nehmen Sie zur Beantwortung dieser Frage an, dass jeder mögliche Satz eine Münze mit Wertigkeit 1 enthält.

88. Aufgabe

Track ε	3
Track AI	3

Formulieren Sie das SSSPP für Graphen als lineares Programm; halten Sie hierfür den Graph $G = (V, E, g)$ allgemein, entwerfen Sie also eine Reduktionsvorschrift, die Instanzen des SSSPP auf lineare Programme abbildet.

Begründen Sie die Korrektheit Ihres Entwurfs und illustrieren Sie ihn anhand des Graphen aus Beispiel 4.1.

89. Aufgabe

	a)	b)	c)
Track ε	[1]	[1]	[1]
Track AI	[1]	[1]	[1]

Wir definieren co-$\mathcal{NP}$ als die Menge all jener Sprachen L, deren Komplemente zur Klasse $\mathcal{NP}$ gehören, also

$$\text{co-}\mathcal{NP} := \{L \in \Sigma^* \mid (\Sigma^* \setminus L) \in \mathcal{NP}\} \ .$$

a) Zeigen Sie: $\mathcal{P} \subseteq \text{co-}\mathcal{NP}$.

b) Zeigen Sie: $NP \neq \text{co-}\mathcal{NP} \implies \mathcal{P} \neq \mathcal{NP}$.

c) Zeigen Sie: Wenn es ein $L \in \mathcal{NPV}$ gibt, sodass $\overline{L} \in NP$, dann gilt $\mathcal{NP} = \text{co-}\mathcal{NP}$.

90. Aufgabe

Track ε	2
Track AI	2

Beweisen Sie durch eine Reduktion von 3KNF-SAT, dass das Problem Clique $\mathcal{NP}$-vollständig ist.

91. Aufgabe

Track ε	[3]
Track AI	[3]

Wir erweitern unseren Katalog $\mathcal{NP}$-vollständiger Probleme um zwei weitere:

- Rucksack: Hier sind die natürlichen Zahlen $a_1, a_2, \ldots, a_k \in \mathbb{N}$ und $b \in \mathbb{N}$ gegeben; es ist zu entscheiden, ob es eine Teilmenge $I \subseteq \{1, 2, \ldots, k\}$ derart gibt, dass $\sum_{i \in I} a_i = b$ gilt.
- Partition: Für die gegebenen natürlichen Zahlen $a_1, a_2, \ldots, a_k \in \mathbb{N}$ ist gefragt, ob es eine Teilmenge $J \subseteq \{1, 2, \ldots, k\}$ gibt, für die $\sum_{i \in J} a_i = \sum_{i \in \{1,2,\ldots,k\} \setminus J} a_i$ gilt.

Beweisen Sie unter der Annahme, dass Rucksack $\mathcal{NP}$-vollständig ist, dass dann auch Partition $\mathcal{NP}$-hart ist.

Es gibt zahlreiche bekannte $\mathcal{NP}$-vollständige Probleme, die sich eignen, um Reduktion unter solchen zu üben. Hier ist eine kleine Auswahl; die genauen Definitionen können Sie in den einschlägigen Quellen finden.

Logik: 3KNF-SAT (auch 3SAT), MAX-SAT

Graphen: Vertex Cover, Edge Cover, Dominating Set, Hamiltonian Path, Hamiltonian Cycle, Traveling Salesperson, Clique, Longest Path, Independent Set

Packen: Partition, Bin Packing, Subset Sum, Set Cover, Knapsack, Rucksack

Sonstige: Pancake Sorting, Bimaru, Sudoku, Mastermind, Tetris(varianten)

Es lohnt sich, sowohl Reduktionen unter den Problemen innerhalb einer „Klasse“ als auch über Klassengrenzen hinweg zu versuchen.