

Übungsblatt 2

Besprechung in 1 Woche

Problem 1-1 Betrachten Sie folgendes Rucksack/Knapsack Problem (Rucksackgröße ist 7) und lösen Sie es mittels Dynamischem Programmieren.

Objekt	1	2	3	4	5
Wert	4	7	3	5	4
Gewicht	5	3	2	2	1

Problem 1-2 Berechnen Sie die "Longest Common Subsequence" von ALGORITHMUS und ALLIGATOR mittels Dynamischem Programmieren.

Problem 1-3 Berechnen Sie die Edit-Distanz zwischen ALGORITHMUS und ALLIGATOR mittels Dynamischem Programmieren. Nehmen Sie folgende Kosten für die Edit-Operationen an: Ändern=3, Löschen=2, Einfügen=5

Wie können Sie die entsprechende Sequenz an Edit-Operationen bestimmen?

Problem 2-1

Betrachten Sie den Graphen mit Knotenmenge $\{1, 2, \dots, 9\}$ und Kantenmenge

$\{(7, 4, 7), (7, 6, 3), (7, 9, 1), (6, 1, 1), (1, 4, 1), (4, 9, 1), (9, 3, 1), (9, 5, 2), (3, 2, 1), (2, 4, 1), (5, 2, 1), (4, 6, 1)\}$

Hier bezeichnet (i, j, c) eine gerichtete Kante von Knoten i nach Knoten j mit Kosten c .

- Malen Sie den Graphen hin
- Zeigen Sie, in welcher Reihenfolge DFS ausgehend von 7 die anderen Knoten besucht; gehen Sie dabei davon aus, dass ausgehende Kanten in Reihenfolge gemäß der Nummer des Zielknotens angeschaut werden (d.h. zum Beispiel Kante $(1, 5)$ wird vor $(1, 8)$ angeschaut, da $5 < 8$).
- Zeigen Sie, in welcher Reihenfolge BFS ausgehend von 7 die anderen Knoten besucht.
- Zeigen Sie den Ablauf von Dijkstra ausgehend von Knoten 7 auf.

Problem 2-2 Entwerfen Sie einen Algorithmus, der auf Graphen mit allgemeinen Kantenkosten für jeden Knoten v bestimmt (d.h. modifizieren Sie einen der in der Vorlesung behandelten Algorithmen):

- $d(v)$ falls $d(v) < \infty$
- dass v auf einem negativen Zyklus liegt

Argumentieren Sie, dass Ihr Algorithmus das Richtige tut (nicht unbedingt formal beweisen).