

Allgemeine Hinweise:

- Die **Hausaufgaben** sollen in Gruppen von je **2-3 Studierenden** aus der **gleichen Kleingruppenübung (Tutorium)** bearbeitet werden. **Namen und Matrikelnummern** der Studierenden sind auf jedes Blatt der Abgabe zu schreiben. **Heften bzw. tackern Sie die Blätter!**
- Die **Nummer der Übungsgruppe** muss **links oben** auf das **erste Blatt** der Abgabe geschrieben werden. Notieren Sie die Gruppennummer gut sichtbar, damit wir besser sortieren können.
- Die Lösungen müssen **bis Mittwoch, den 15.07.2015 um 12:15 Uhr** in den entsprechenden Übungskasten eingeworfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55). Alternativ können Sie die Lösungen auch vor der Abgabefrist direkt bei Ihrer Tutorin/Ihrem Tutor abgeben.
- In Aufgaben, bei denen Sie Algorithmen implementieren sollen, dürfen Sie Ihre Lösung als Pseudo-Code abgeben. Abgaben in verbreiteten imperativen Sprachen wie Java oder C++ sind ebenfalls erlaubt.
- Aufgaben, die mit einem * markiert sind, sind Sonderaufgaben mit erhöhtem Schwierigkeitsgrad. Sie tragen nicht zur Summe der erreichbaren Punkte bei, die für die Klausurzulassung relevant ist, jedoch werden Ihnen die in solchen Aufgaben erreichten Punkte ganz normal gutgeschrieben.

Tutoraufgabe 1 (Longest Common Subsequence):

Bestimmen Sie die *längste gemeinsame Teilsequenz* der Sequenzen AACHEN und ANDERNACH. Benutzen Sie hierfür den in der Vorlesung vorgestellten Algorithmus mit dynamischer Programmierung und füllen Sie die folgende Tabelle aus.

	∅	A	N	D	E	R	N	A	C	H
∅										
A										
A										
C										
H										
E										
N										

Tutoraufgabe 2 (Dynamische Programmierung):

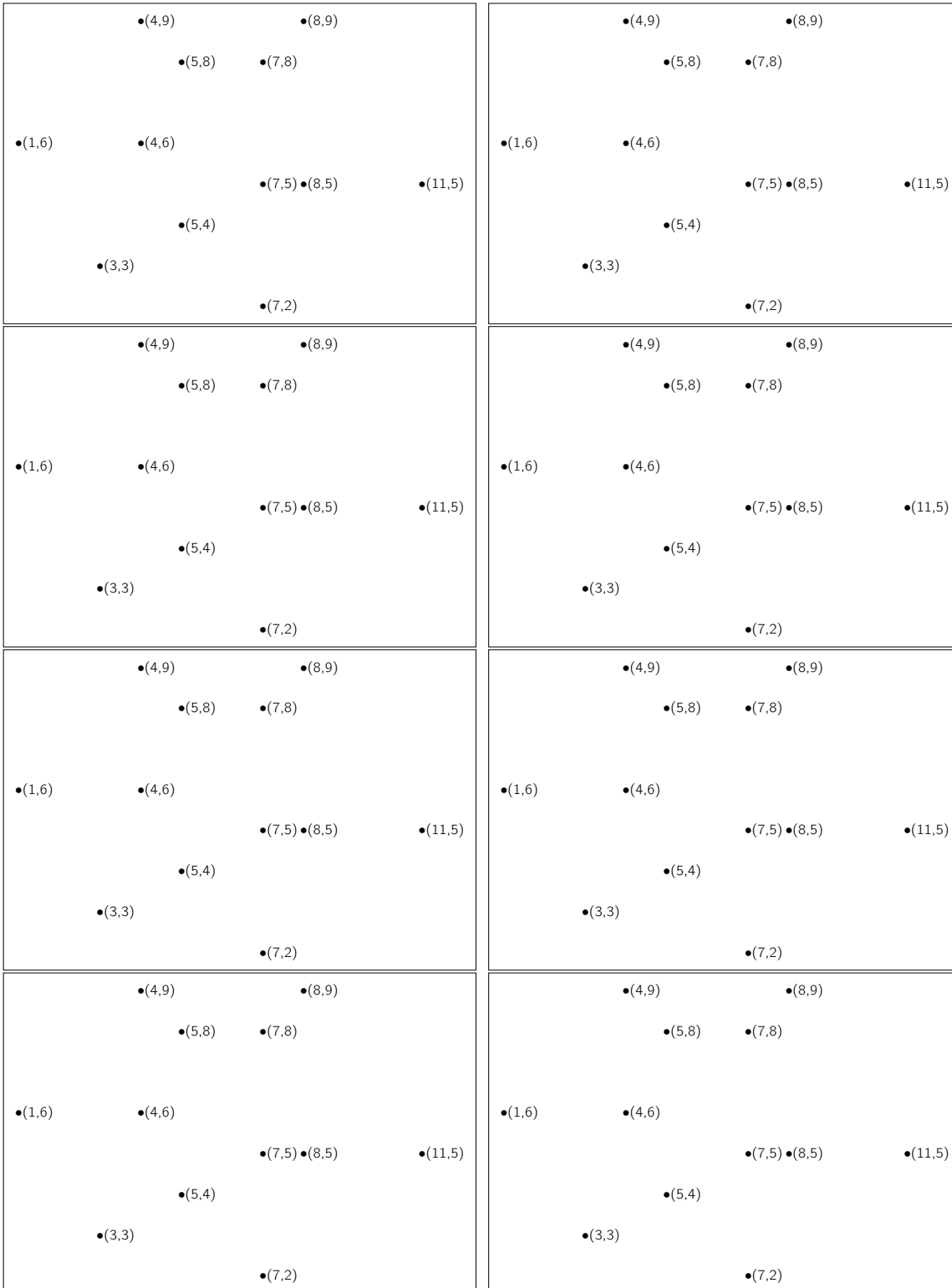
- a) Schreiben Sie eine Funktion, welche die Länge der Longest Common Subsequence (LCS) für zwei gegebene Zeichensequenzen berechnet. Die Zeichensequenzen seien dabei der Einfachheit halber `int` Arrays `seq1` und `seq2` mit den Längen `l1` and `l2`. Demnach soll Ihre Funktion die folgende Signatur haben:

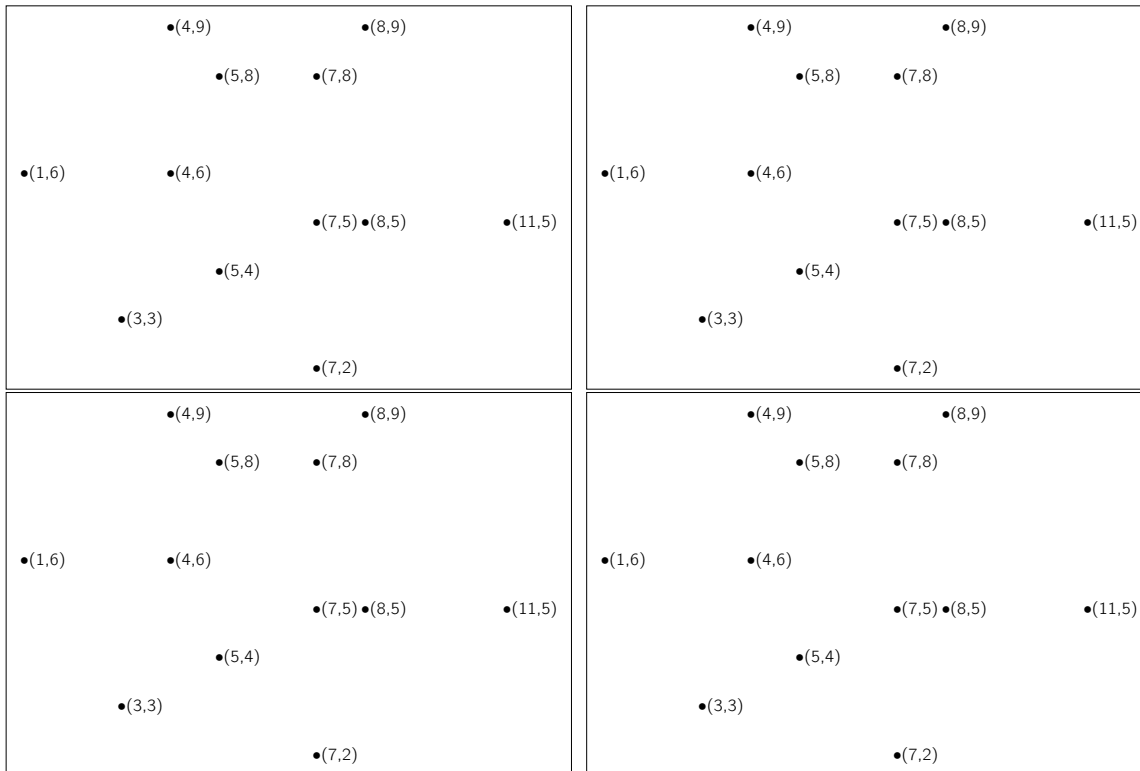
```
int lcs(int seq1[], int l1, int seq2[], int l2)
```

- b) Betrachten Sie folgendes Szenario. Ihnen wird eine Klausur gestellt, welche n Aufgaben umfasst. Jede dieser Aufgaben hat eine Punktzahl p_i und eine von Ihnen geschätzte Zeit t_i , die Sie zum Lösen der Aufgabe benötigen ($1 \leq i \leq n$). Geben Sie die maximale Punktzahl, welche Sie in T Zeiteinheiten erreichen können, als Rekursionsgleichung an (inklusive der passenden Argumente).

Tutoraufgabe 3 (Graham-Scan):

Berechnen Sie die konvexe Hülle der folgenden Punktmenge. Benutzen Sie dafür *Grahams Scan* wie *in der Vorlesung* vorgestellt und geben Sie die Teilschritte *nach jeder Iteration* (also nach jedem neu hinzugefügten Punkt) an. Umkreisen Sie die Punkte, die vom Algorithmus in der Iterationsschleife nicht betrachtet werden.





Aufgabe 4 (Rucksackproblem):

(4 Punkte)

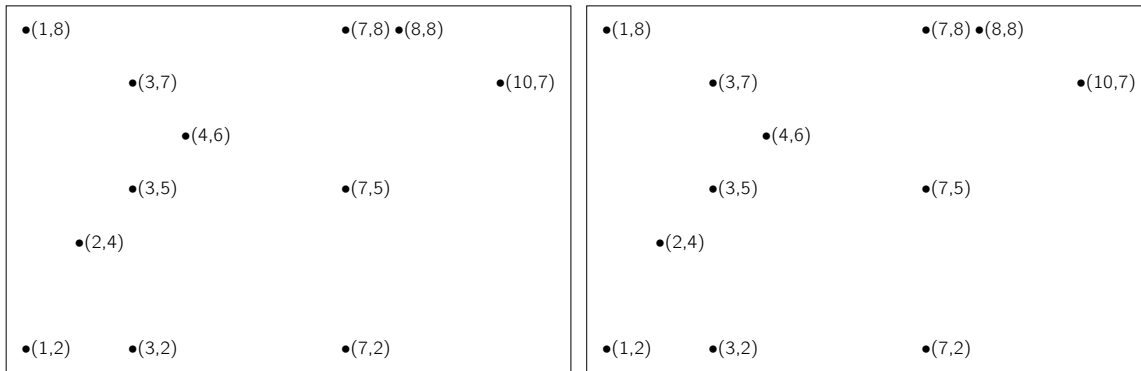
Gegeben sei ein Rucksack mit *maximaler Tragkraft* 10 sowie 5 *Gegenstände*. Der *i*-te Gegenstand soll hierbei ein Gewicht von w_i und einen Wert von c_i haben. Bestimmen Sie mit Hilfe des in der Vorlesung vorgestellten Algorithmus zum Lösen des Rucksackproblems mit dynamischer Programmierung den maximalen Gesamtwert der Gegenstände, die der Rucksack tragen kann (das Gesamtgewicht der mitgeführten Gegenstände übersteigt nicht die Tragkraft des Rucksacks). Die *Gewichte* seien dabei $w_1 = 5$, $w_2 = 2$, $w_3 = 1$, $w_4 = 3$ und $w_5 = 4$ und die *Werte* $c_1 = 9$, $c_2 = 8$, $c_3 = 6$, $c_4 = 7$ und $c_5 = 5$. Geben Sie zudem die vom Algorithmus bestimmte Tabelle C und die mitzunehmenden Gegenstände an.

	0	1	2	3	4	5	6	7	8	9	10
0											
1											
2											
3											
4											
5											

Aufgabe 5 (Dynamische Programmierung):

(4 + 4 + 3 + 3 = 14 Punkte)

Gegeben sei ein $(n \times m)$ -Schachbrett (d. h. n Zeilen, m Spalten) mit $n, m \geq 4$. Auf jedem Feld (i, j) (d. h. Zeile i , Spalte j) liegt eine Münze mit dem Wert w_{ij} .



Aufgabe 7 (Klausur):

(20* Punkte)

a)* Entwerfen Sie eine Klausur (Aufgabenstellung und Musterlösung) für das Fach Datenstrukturen und Algorithmen mit einem zeitlichen Umfang von 120 Minuten. Die Klausur soll die wesentlichen Konzepte der Vorlesung abprüfen, einen angemessenen Schwierigkeitsgrad haben sowie klar und verständlich formuliert sein.

Beachten Sie, dass Plagiarismus zu einer Bewertung mit 0 Punkten führt.