

1. (2 Punkte) Gegeben sei die Tupelliste $a = [(3,5), (7,3), (1,2), (10,1)]$. Erzeuge mit dem lambda-Operator zwei Listen b1 und b2. b1 ist sortiert nach der Summe der Komponenten, b2 ist absteigend nach der 2. Komponente sortiert.

Lösung:

```
b1 = sorted(a, key = lambda x: x[0]+x[1])
b2 = sorted(a, key = lambda x: x[1], reverse=True)
```

2. (2 Punkte) Gegeben sei die Tupelliste $a = [(3,5,4), (7,3,2), (1,2,1), (10,1,6)]$. Erzeuge mit dem lambda-Operator zwei Listen b1 und b2. b1 ist sortiert nach dem Produkt der drei Komponenten, b2 ist absteigend nach der letzten Komponente sortiert.

Lösung:

```
b1 = sorted(a, key = lambda x: x[0]*x[1]*x[2])
b2 = sorted(a, key = lambda x: x[2], reverse=True)
```

3. (2 Punkte) Gegeben sei die Liste $a = [2301, 5401, 1303, 6403, 4302, 2402]$.

Sortiere die Liste mit einer Zeile nach den 100er-Zahlen. Das Ergebnis sieht dann so aus:

2301 1303 4302 5401 6403 2402.

Lösung:

```
a.sort(key = lambda x: (x%1000)//100)
```

4. (2 Punkte) Gegeben sei eine Liste a mit positiven ganzen Zahlen. Sortiere die Liste abwärts nach der Quersumme der Zahlen.

Lösung:

```
a.sort(key = lambda x: sum([int(k) for k in str(x)]), reverse=True)
```

5. (3 Punkte) Gegeben sei das Rucksackproblem mit folgenden Eingaben:

```
wert = [1,11,13,9]
gewicht = [2,6,8,4]
kapazitaet = 10
```

Benenne die im Unterricht besprochenen drei Greedy-Strategien und ermittle die Indizes der genommenen Items, den Wert und das Gewicht des Rucksacks für jeden der drei Fälle.

Lösung:

Die wertvollsten zuerst: Items = 0 2 , Wert = 14 , Gewicht = 10

Die leichtesten zuerst: Items = 0 3 , Wert = 10 , Gewicht = 6

Die wertdichtesten zuerst: Items = 1 3 , Wert = 20 , Gewicht = 10