

1. Schreiben sie einen Pseudocode, der testet, ob eine gegebene Tupel von Zahlen (a_0, \dots, a_{n-1}) absteigend (also beginnend mit dem größten Wert) sortiert ist.
2. Nehmen Sie an, sie hätten das Programm aus der vorigen Aufgabe zur Bestimmung des Maximums zur Verfügung, sodass es ihnen den Wert und die Stelle des maximalen Elements in einer gegebenen Tupel liefert, i.e.
 $a_n, n = \text{Maximum}(\text{Tupel})$.
 Schreibe Sie einen Pseudocode, der dieses Programm nutzt, um ein Tupel zu sortieren.
3. Wie oft müssen Sie die Tupel durchlaufen?

Solution:

1. Eingabe Tupel der Länge n

```

Wert = Tupel[0]
Für alle i in [1, n):
    Falls Wert < Tupel[i]:
        Ausgabe: Das Tupel ist nicht absteigend sortiert.
    Ende
    Wert = Tupel[i]
Ausgabe: Das Tupel ist absteigend sortiert.
Ende

```

2. Eingabe Tupel1 der Länge n
 Leeres Tupel2 der Länge n anlegen

```

Solange Tupel1 nicht leer ist:
     $a_n, n = \text{Maximum}(\text{Tupel1})$ 
    Lösche Element  $n$  aus Tupel1
    Füge Wert  $a_n$  am Ende von Tupel2 hinzu
Ausgabe: Tupel2

```

3. Maximum läuft in der Iteration k ein Tupel der Länge k ab. Das machen wir für Tupel der Länge $n, n-1, \dots, 1$. Damit erhalten wir $n(n+1)/2$ durchläufe.