

## Übungsblatt Nr. 7

### Aufgabe 7.1

Dokumentation der Laufzeit in Sekunden für die Sortierung eines Arrays mit 500 Zahlen – erzeugt über die Methode `generate_random_numbers`.

Bubblesort	Insertionsort	Quicksort
0.12563	0.03512	0.00442

Dokumentation der UI-Testfälle für die Sortierung:

Funktionalität	Eingabe	Verhalten des Programms
<b>Wahl zur Eingabe der Liste</b>		In der Konsole wird eine Benutzereingabe angezeigt, in der der Benutzer seine Wahl angeben kann (0 oder 1).
Wahl zur Eingabe der Liste	$\emptyset$	Nutzer erhält den Hinweis, dass seine Eingabe leer war und wird zu einer neuen Eingabe aufgefordert. Zudem wird ihm mitgeteilt, dass er durch einen KeyboardInterrupt das Programm beenden kann.
Wahl zur Eingabe der Liste	Literal enthält Element $\notin \{0, 1\}$ Bsp. <b>a</b>	Nutzer erhält den Hinweis, dass seine Eingabe nicht konform war und wird zu einer neuen Eingabe aufgefordert.
Wahl zur Eingabe der Liste	0	Der Computer erzeugt ein Array mit 200 Einträgen im Bereich [1 ... 20]. Dieses wird dem Nutzer angezeigt, und er wird zur Eingabe der gewünschten Algorithmen aufgefordert.
Wahl zur Eingabe der Liste	1	Der Nutzer wird zur Eingabe einer Liste zu sortierender Zahlen aufgefordert.
Wahl zur Eingabe der Liste	Keyboard Interrupt	Das Programm wird mit einer Auskunft zum KeyboardInterrupt beendet.
<b>Eingabe der zu sortierenden Liste</b>		In der Konsole wird eine Benutzereingabe mit Beispiel angezeigt, in der der Benutzer seine zu sortierende Liste eingeben kann.
Eingabe der zu sortierenden Liste	$\emptyset$	Nutzer erhält den Hinweis, dass seine Eingabe leer war und wird zu einer neuen Eingabe aufgefordert. Zudem wird ihm mitgeteilt, dass er durch einen KeyboardInterrupt das Programm beenden kann.
Eingabe der zu sortierenden Liste	Literal enthält Elemente die nicht in float umgewandelt werden können	Nutzer erhält den Hinweis, dass seine Eingabe nicht konform war und wird zu einer neuen Eingabe aufgefordert (ValueError).
Eingabe der zu sortierenden Liste	-1_+1_999.0	Die Eingabe wird in ein Array umgewandelt. Der Nutzer wird danach zur Eingabe der anzuwendenden Algorithmen aufgefordert.
Eingabe der zu sortierenden Liste	0__1	Aufgrund der überschüssigen Leerzeichen erhält der Nutzer den Hinweis, dass seine Eingabe nicht konform war (ValueError). Er wird zu einer neuen Eingabe aufgefordert.
Eingabe der zu sortierenden Liste	Keyboard Interrupt	Das Programm wird mit einer Auskunft zum KeyboardInterrupt beendet.

<b>Eingabe der anzuwendenden Algorithmen</b>		In der Konsole wird eine Benutzereingabe mit Beispiel angezeigt, in der der Benutzer seine zu Auswahl an anzuwendenden Algorithmen eingeben kann.
Eingabe der anzuwendenden Algorithmen	$\emptyset$	Nutzer erhält den Hinweis, dass seine Eingabe leer war und wird zu einer neuen Eingabe aufgefordert. Zudem wird ihm mitgeteilt, dass er durch einen KeyboardInterrupt das Programm beenden kann.
Eingabe der anzuwendenden Algorithmen	0_1_2_3	Die Eingabe ist korrekt. Der Nutzer wird aufgefordert, mit <code>&lt;Enter&gt;</code> die einzelnen Algorithmen nacheinander ablaufen zu lassen.
Eingabe der anzuwendenden Algorithmen	0__1	Aufgrund der überschüssigen Leerzeichen erhält der Nutzer den Hinweis, dass seine Eingabe nicht konform war. Er wird zu einer neuen Eingabe aufgefordert.
Eingabe der anzuwendenden Algorithmen	Literal enthält Element $\notin \{0, 1, 2, 3\}$ Bsp. 0_;	Nutzer erhält den Hinweis, dass seine Eingabe leer war und wird zu einer neuen Eingabe aufgefordert.
Eingabe der anzuwendenden Algorithmen	0_0	Die Eingabe ist korrekt (Mehrfachnennung ist möglich, führt aber nicht dazu, dass der entsprechende Algorithmus mehrfach ausgeführt wird). Der Nutzer wird aufgefordert, mit <code>&lt;Enter&gt;</code> die einzelnen Algorithmen nacheinander ablaufen zu lassen. Bubblesort wird nur einmal ausgeführt.
Eingabe der anzuwendenden Algorithmen	Keyboard Interrupt	Das Programm wird mit einer Auskunft zum KeyboardInterrupt beendet.
<b>Anzeige der Ergebnisse</b>	korrekte Eingabe bei Algorithmenauswahl	Die jeweiligen Algorithmen werden durchgeführt und deren Ergebnisse in der Konsole ausgegeben.
Anzeige der Ergebnisse	korrekte Eingabe bei Algorithmenauswahl: 0	Bubblesort wird durchgeführt. Das eingegebene Array wird angezeigt. Das sortierte Array wird ebenfalls angezeigt. Die Laufzeit wird ausgegeben. Das sortierte Ergebnis von <code>np.sort</code> wird angezeigt. Es gibt einen Hinweis, ob die beiden sortierten Arrays übereinstimmen.
Anzeige der Ergebnisse	korrekte Eingabe bei Algorithmenauswahl: 1	Insertionsort wird durchgeführt. Das eingegebene Array wird angezeigt. Das sortierte Array wird ebenfalls angezeigt. Die Laufzeit wird ausgegeben. Das sortierte Ergebnis von <code>np.sort</code> wird angezeigt. Es gibt einen Hinweis, ob die beiden sortierten Arrays übereinstimmen.
Anzeige der Ergebnisse	korrekte Eingabe bei Algorithmenauswahl: 2	Quicksort wird durchgeführt. Das eingegebene Array wird angezeigt. Das sortierte Array wird ebenfalls angezeigt. Die Laufzeit wird ausgegeben. Das sortierte Ergebnis von <code>np.sort</code> wird angezeigt. Es gibt einen Hinweis, ob die beiden sortierten Arrays übereinstimmen.
Anzeige der Ergebnisse	korrekte Eingabe bei Algorithmenauswahl: 3	Bubblesort wird für zehn zufällig gewählten Zahlen durchgeführt. Das eingegebene Array wird in der Konsole angezeigt. Das sortierte Array wird ebenfalls dort angezeigt. Die Laufzeit wird ausgegeben. Das sortierte Ergebnis von <code>np.sort</code> wird angezeigt. Es gibt einen Hinweis, ob die beiden sortierten Arrays übereinstimmen. Das Sortiervverfahren wird zudem durch wandernde Balken in einem Diagramm visualisiert.

Screenshots von Aufrufen von `tests.py` zum Testen der Sortieralgorithmen:

```
[larspetersen@lpmhp 07_Uebung]$ python3 tests.py
test_bubble_sort (__main__.TestSortAlgorithms) ...
Eingabe:
[-25 11 -50 35 -47 5 32 11 -12 -12 1 -34 -41 7 16 -31 -34 30
-21 -32 -31 9 10 -50 35 7 31 33 16 34 4 -5 -14 -30 -38 -15
7 -22 -38 -27]
Ausgabe:
[-50 -50 -47 -41 -38 -38 -34 -34 -32 -31 -31 -30 -27 -25 -22 -21 -15 -14
-12 -12 -5 1 4 5 7 7 7 9 10 11 11 16 16 30 31 32
33 34 35 35]
ok
test_insertion_sort (__main__.TestSortAlgorithms) ...
Eingabe:
[-49 -1 35 -41 26 32 -22 -46 -11 20 32 -5 3 36 34 -16 50 -14
43 2 -31 -39 -14 25 46 30 1 -4 -35 14 2 9 -13 -21 -16 17
-40 -39 9 -37 -45 15 33 -6 -18 36 -47 -17 -27]
Ausgabe:
[-49 -47 -46 -45 -41 -40 -39 -39 -37 -35 -31 -27 -22 -21 -18 -17 -16 -16
-14 -14 -13 -11 -6 -5 -4 -1 1 2 2 3 9 9 14 15 17 20
25 26 30 32 32 33 34 35 36 36 43 46 50]
ok
test_quick_sort (__main__.TestSortAlgorithms) ...
Eingabe:
[-48 -49 -15 -33 24 -31 -12 -29 -12 -18 -50 -41 -43 -31 8 8 46 4
15 -18 -13 -22 -20 -3 15 29 -15 19 7 42 -24 3 19 24 24 -21
-27 -26 -12 13]
Ausgabe:
[-50 -49 -48 -43 -41 -33 -31 -31 -29 -27 -26 -24 -22 -21 -20 -18 -18 -15
-15 -13 -12 -12 -12 -3 3 4 7 8 8 13 15 15 19 19 24 24
24 29 42 46]
ok
-----
Ran 3 tests in 0.007s

OK
```

```
[larspetersen@lpmhp 07_Uebung]$ python3 tests.py
test_bubble_sort (__main__.TestSortAlgorithms) ...
Eingabe:
[-25 11 -50 35 -47 5 32 11 -12 -12 1 -34 -41 7 16 -31 -34 30
-21 -32 -31 9 10 -50 35 7 31 33 16 34 4 -5 -14 -30 -38 -15
7 -22 -38 -27]
Ausgabe:
[-50 -50 -47 -41 -38 -38 -34 -34 -32 -31 -31 -30 -27 -25 -22 -21 -15 -14
-12 -12 -5 1 4 5 7 7 7 9 10 11 11 16 16 30 31 32
33 34 35 35]
ok
test_insertion_sort (__main__.TestSortAlgorithms) ...
Eingabe:
[-49 -1 35 -41 26 32 -22 -46 -11 20 32 -5 3 36 34 -16 50 -14
43 2 -31 -39 -14 25 46 30 1 -4 -35 14 2 9 -13 -21 -16 17
-40 -39 9 -37 -45 15 33 -6 -18 36 -47 -17 -27]
Ausgabe:
[-49 -47 -46 -45 -41 -40 -39 -39 -37 -35 -31 -27 -22 -21 -18 -17 -16 -16
-14 -14 -13 -11 -6 -5 -4 -1 1 2 2 3 9 9 14 15 17 20
25 26 30 32 32 33 34 35 36 36 43 46 50]
ok
test_quick_sort (__main__.TestSortAlgorithms) ...
Eingabe:
[-48 -49 -15 -33 24 -31 -12 -29 -12 -18 -50 -41 -43 -31 8 8 46 4
15 -18 -13 -22 -20 -3 15 29 -15 19 7 42 -24 3 19 24 24 -21
-27 -26 -12 13]
Ausgabe:
[-50 -49 -48 -43 -41 -33 -31 -31 -29 -27 -26 -24 -22 -21 -20 -18 -18 -15
-15 -13 -12 -12 -12 -3 3 4 7 8 8 13 15 15 19 19 24 24
24 29 42 46]
ok
-----
Ran 3 tests in 0.007s

OK
```