Aufgabe 1

Sortiere die Liste `[3, 5, 1, 6]` in-place mit Gnomesort, indem du jeden elementaren Sortierschritt in einer Zeile protokollierst und jeweils die Position des Gartenzwergs durch Unterstreichen kennzeichnest.

Aufgabe 2

Vervollständige die Python-Funktion `gnomesort(L)`, welche die Liste `L` aufsteigend sortiert. Rücke jeweils 2 Zeichen ein.

```
def gnomesort(L):
```

Aufgabe 3

Diskutiere die Laufzeitkomplexität von Gnomesort.

Aufgabe 4

Sortiere die Liste `[7, 9, 1, 3, 5]` in-place mit Selectionsort, indem du die wesentlichen Schritte des Verfahrens zeilenweise darstellst und jeweils die Anzahl der Vergleiche und die Anzahl der Vertauschungen protokollierst.

Aufgabe 5

Vervollständige in der Python-Funktion `selectionsort(L)`, welche die Liste `L` mit Selectionsort aufsteigend sortiert, die mit den Fragezeichen markierten Zeilen.

```
def selectionsort(L):
?
    for i in range(0,n-1):
        minpos=i
        for i in range(i+1,n):
            if A[j]<A[minpos]:
?
        A[minpos],A[i]=A[i],A[minpos]
```

Aufgabe 6

Diskutiere die Laufzeitkomplexität von Selectionsort.

Aufgabe 7

Sortiere die Liste `[7, 9, 1, 3, 5]` in-place mit Insertionsort, indem du die wesentlichen Schritte des Verfahrens zeilenweise darstellst und jeweils die Anzahl der Vergleiche und die Anzahl verschobener Elemente protokollierst.

Aufgabe 8

Vervollständige in der Python-Funktion `insertionsort(L)`, welche die Liste `L` mit Insertionsort aufsteigend sortiert, die mit den Fragezeichen markierten Zeilen.

```
def insertionsort(L):
    n = len(A):
?
    x = A[i]
    j = i - 1
    while (j>=0 and A[j]>x):
        A[j+1] = A[j]
?
    A[j+1] = x
```

Aufgabe 9

Diskutiere die Laufzeitkomplexität von Insertionsort

Aufgabe 10

Sortiere die Liste `[9, 5, 1, 8, 7, 4, 6]` mit Quicksort, indem du die listenverändernden Schritte des Verfahrens zeilenweise darstellst

Aufgabe 11

Diskutiere die Laufzeitkomplexität von Quicksort

Aufgabe 12

Für welche Art von Listen sortiert Quicksort schlecht und was können wir dagegen unternehmen?

Aufgabe 13

Welcher Sortieralgorithmus wird vom folgenden Python-Programm implementiert?

```
def sort(A):
    n = len(A)
    for i in range(0, n-1):
        k = i
        for j in range(i+1, n):
            if A[j] < A[k]:
                k = j
        A[k], A[i] = A[i], A[k]
```

Aufgabe 14

Welcher Sortieralgorithmus wird vom folgenden Python-Programm implementiert?

```
def sort(L):
    n = len(L)
    i = 0
    while i < n:
        if i == 0:
            i = i + 1
        elif L[i-1] > L[i]:
            L[i], L[i-1] = L[i-1], L[i]
            i = i - 1
        else:
            i = i + 1
```

Aufgabe 15

Welcher Sortieralgorithmus wird vom folgenden Python-Programm implementiert?

```
def sort(A):
    for i in range(1, len(A)):
        x = A[i]
        j = i-1
        while(j >= 0 and A[j] > x):
            A[j+1] = A[j]
            j = j-1
        A[j+1] = x
```

Aufgabe 16

Welcher Sortieralgorithmus wird vom folgenden Python-Programm implementiert?

```
def sort(A):
    sort_helper(A, 0, len(A))

def helper(A, a, b):
    if a < b:
        m = function(A, a, b)
        helper(A, a, m)
        helper(A, m+1, b)

def function(A, a, b):
    p = A[b-1]
    i = a-1
    for j in range(a, b-1):
        if A[j] <= p
            i += 1
            A[i], A[j] = A[j], A[i]
    A[i+1], A[b-1] = A[b-1], A[i+1]
    return i+1
```

Aufgabe 17

Die Tabellen zeigen, wie drei verschiedene Sortieralgorithmen den Zustand einer Liste verändern, wobei nicht immer fertig sortiert wird.

7	2	4	3	8	5
2	7	4	3	8	5
2	3	4	7	8	5
2	3	4	5	8	7
2	3	4	5	7	8

7	2	4	3	8	5
2	7	4	3	8	5
2	4	7	3	8	5
2	4	3	7	8	5
2	4	3	5	8	7

7	2	4	3	8	5
2	7	4	3	8	5
2	4	7	3	8	5
2	4	3	7	8	5
2	3	4	7	8	5

Um welchen der folgenden Algorithmen handelt es sich jeweils?

- Gnomesort
- Selectionsort
- Insertionsort
- Quicksort

Aufgabe 18

Die Tabellen zeigen, wie drei verschiedene Sortieralgorithmen den Zustand einer Liste verändern, wobei nicht immer fertig sortiert wird.

4	3	8	1	9	5
3	4	8	1	9	5
3	4	1	8	9	5
3	1	4	8	9	5
1	3	4	8	9	5

4	3	8	1	9	5
1	3	8	4	9	5
1	3	4	8	9	5
1	3	4	5	9	8
1	3	4	5	8	9

4	3	8	1	9	5
3	4	8	1	9	5
1	3	4	8	9	5
1	3	4	5	8	9

Um welchen der folgenden Algorithmen handelt es sich jeweils?

- Gnomesort
- Selectionsort
- Insertionsort
- Quicksort

Aufgabe 19

Eine Implementierung von Selectionsort liegt in $O(n^2)$ und benötigt etwa 6 Sekunden, um 10^4 Datensätze zu sortieren.

Schätze, wie lange das Programm für das Sortieren von 10^5 Datensätzen braucht.

Aufgabe 20

Eine Implementierung von Quicksort liegt in $O(n \log n)$ und benötigt etwa 6 Sekunden, um 10^4 Datensätze in zufälliger Reihenfolge zu sortieren.

Schätze, wie lange das Programm für das Sortieren von 10^5 Datensätzen in zufälliger Reihenfolge braucht.

Hinweis: Die Wahl der Logarithmenbasis hat keinen Einfluss auf das Resultat.