

1. (3 Punkte) Sortiere die Zahlenfolge mit SelectionSort. Schreibe für die ersten drei Durchgänge je eine Zeile.

13 4 92 42 11 7 12

2. (4 Punkte) Notiere den Code, der für den SelectionSort-Algorithmus fehlt. Gib auch die Stufe der Einrückung an.

```
def selection_sort(a):           #E0
    for i in range(len(a)-1):     #E1
        pos = i                  #E2
        min = a[i]               #E2
                                ???(1)
        if a[j] < min:           #E3
            pos = j              #E4
            min = a[j]           #E4
                                ???(2)
```

3. (3 Punkte) Sortiere die Zahlenfolge mit BubbleSort. Schreibe für die ersten drei Durchgänge je eine Zeile.

2 22 14 25 1 13 9

4. (4 Punkte) Notiere den Code, der für den BubbleSort-Algorithmus fehlt. Gib auch die Stufe der Einrückung an.

```
def bubble_sort(a):  
    while getauscht:  
        getauscht = False  
        for i in range(len(a)-1):  
            a[i], a[i+1] = a[i+1], a[i]  
            getauscht = True
```

```
#E0  
??? (1)  
#E1  
#E2  
#E2  
??? (2)  
#E4  
#E4
```

5. (5 Punkte) Die Liste `a = [24, 4, 17, 88, 42, 12, 7]` wird mit dem rekursiven `mergeSort`-Algorithmus aus dem Unterricht sortiert.

- Wieviel mal wird `merge` aufgerufen?
- Wieviel mal wird `mergeSort` aufgerufen? (der erste Aufruf zählt mit).

Bei jedem Aufruf von `merge` wird eine Liste zurückgegeben. Notiere die Listen in der Reihenfolge, in der sie zurückgegeben werden.

6. (5 Punkte) Die Liste 10 5 67 18 3 22 7 wird mit `quickSort` sortiert. Schreibe die erste drei Protokollzeilen.

7. (5 Punkte) Die Liste 22 41 43 7 42 19 wird mit quickSort sortiert. Schreibe die ersten drei Protokollzeilen.

8. (2 Punkte) Der nächste Quicksort-Durchgang bearbeitet die Liste von 0-3.

15 8 9 16 18 28 22 38 26

Schreibe die Protokollzeilen vor und nach dem Durchgang.

9. (2 Punkte) Quicksort erhält die Liste zur Sortierung. Schreibe die Protokollzeilen vor und nach dem ersten Durchgang.

15 26 22 18 16 28 9 38 8

10. (2 Punkte) Mache aus der Liste einen Heap nach dem Verfahren aus dem Unterricht.

12 6 3 17 42 5 25 38 9 67 54 1 81

11. (4 Punkte) Eine Liste wird mit HeapSort sortiert. Die Zeile zeigt die in einen Heap umgewandelte Liste. Füge für die beiden folgenden Reorganisationen je eine Zeile hinzu.

1 6 4 17 28 33 20 92

12. (2 Punkte) Welche Komplexität hat die Laufzeit von Quicksort im best, worst und average-case? In welcher Komplexitätsklasse ist der zusätzliche Platzbedarf?