

1. (2 Punkte) Sortiere die Zahlenfolge mit SelectionSort. Schreibe für die ersten drei Durchgänge je eine Zeile.

13 4 92 42 11 7 12

2. (2 Punkte) Sortiere die Zahlenfolge mit BubbleSort. Schreibe für die ersten drei Durchgänge je eine Zeile.

2 22 14 25 1 13 9

3. (3 Punkte) In unserer Implementierung des SelectionSort-Algorithmus wurde ein print-Statement eingefügt. Welche Zahlenfolge wird bei dem angegebenen Aufruf von SelectionSort ausgegeben? (Der print-Parameter end=' ' sorgt dafür, dass die Zahlen nicht in einer neuen Zeile, sondern mit blank getrennt ausgegeben werden.)

```
def selection_sort(a):
    for i in range(len(a)-1):
        best = i
        best_val = a[i]
        for j in range(i+1,len(a)):
            if a[j] < best_val:
                best = j
                print(best,end= ' ')
                best_val = a[j]
        a[best],a[i] = a[i],a[best]
```

```
selection_sort([7,3,1,8,5,12,5,4])
```

4. (2 Punkte) In unserer Implementierung des Bubblesort-Algorithmus wurden Variablennamen geändert. Dabei haben sich Fehler eingeschlichen. Gib die Zeilennummern mit den Fehlern an und korrigiere die fehlerhaften Stellen. (Wir zählen Zeilennummern mit 1 beginnend).

```
def bubble_sort(alpha):  
    beta = True  
    while beta:  
        beta = False  
        for delta in range(len(alpha)-1):  
            if alpha[gamma] > alpha[gamma+1]:  
                alpha[gamma], alpha[gamma+1] = alpha[gamma+1], alpha[gamma]  
            beta = True
```

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 `22 41 43 7 42 19` wird mit `quickSort` sortiert. Schreibe die ersten drei Protokollzeilen.

7. (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.

8. (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

9. (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

10. (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

11. (2 Punkte) Welche Komplexität hat die Laufzeit von BubbleSort im best, worst und average-case? In welcher Komplexitätsklasse ist der zusätzliche Platzbedarf?