

# Lernnachweis B1F

Die Fähigkeit, Algorithmen in funktionale Teilstücke aufzuteilen, wurde durch meine intensive Auseinandersetzung mit verschiedenen Programmierparadigmen gestärkt. Ein herausragendes Beispiel ist die Zerlegung des Merge-Sort-Algorithmus in funktionale Module.

Der Merge-Sort-Algorithmus kann in folgende funktionale Teile aufgeteilt werden:

Teilung (Split): Die Eingabesequenz wird in zwei Hälften geteilt.

Rekursive Sortierung: Beide Hälften werden separat sortiert, indem der Merge-Sort-Algorithmus auf sie angewendet wird.

Vereinigung (Merge): Die sortierten Hälften werden zusammengeführt.

Codebeispiel:

```
def merge_sort(arr):
```

```
    if len(arr) <= 1:
```

```
        return arr
```

```
    # Teilung
```

```
    mid = len(arr) // 2
```

```
    left_half = arr[:mid]
```

```
    right_half = arr[mid:]
```

```
    # Rekursive Sortierung
```

```
    left_half = merge_sort(left_half)
```

```
    right_half = merge_sort(right_half)
```

```
    # Vereinigung
```

```
    return merge(left_half, right_half)
```

```
def merge(left, right):  
    result = []  
    i = j = 0  
  
    while i < len(left) and j < len(right):  
        if left[i] < right[j]:  
            result.append(left[i])  
            i += 1  
        else:  
            result.append(right[j])  
            j += 1  
  
    result.extend(left[i:])  
    result.extend(right[j:])  
    return result
```

# Beispielaufruf

```
arr = [38, 27, 43, 3, 9, 82, 10]  
sorted_arr = merge_sort(arr)  
print(f'Sortierte Liste: {sorted_arr}')
```

Reflexion:

Das Zerlegen von Algorithmen in funktionale Teile erfordert ein tiefes Verständnis der Problemstruktur und der möglichen Modularisierung. Die Reflexion über die Wahl der funktionalen Module und deren Interaktion ist entscheidend für die Effizienz und Lesbarkeit des Codes.

Die Diskussionen in der Entwicklergemeinschaft über verschiedene Ansätze zur Zerlegung von Algorithmen förderten meine kreativen Lösungsansätze und halfen mir, verschiedene Perspektiven zu verstehen. Die Anwendung dieser Fähigkeit in realen Projekten trug dazu bei, meine analytischen und problemlösenden Fähigkeiten zu stärken.

### Zukünftige Schritte:

Um meine Fähigkeiten weiter zu vertiefen, plane ich, an komplexeren Algorithmen zu arbeiten und verschiedene Wege der funktionalen Zerlegung zu erforschen. Die kontinuierliche Anwendung und Erweiterung dieser Kompetenz werden meine Fähigkeiten in der Entwicklung effizienter und gut strukturierter Algorithmen weiter stärken.