

# Lernnachweis B1G

Die Fähigkeit, einen Algorithmus zu erklären, wurde durch meine intensive Auseinandersetzung mit verschiedenen Algorithmen gestärkt. Ein herausragendes Beispiel ist der Dijkstra-Algorithmus zur Suche des kürzesten Pfades in einem gewichteten Graphen.

Der Dijkstra-Algorithmus würde ich wie folgt erklären:

1. Initialisiere alle Knoten mit unendlichem Abstand, außer dem Startknoten mit Abstand 0.
2. Wähle den Knoten mit dem geringsten bekannten Abstand als aktuellen Knoten.
3. Aktualisiere die Abstände der benachbarten Knoten durch den aktuellen Knoten.
4. Markiere den aktuellen Knoten als besucht.
5. Wiederhole die Schritte 2-4, bis alle Knoten besucht sind.

Codebeispiel:

```
graph = {  
    'A': {'B': 1, 'C': 4},  
    'B': {'A': 1, 'C': 2, 'D': 5},  
    'C': {'A': 4, 'B': 2, 'D': 1},  
    'D': {'B': 5, 'C': 1}  
}  
  
def dijkstra(graph, start):  
    # Implementierung des Algorithmus  
  
    start_node = 'A'  
    shortest_paths = dijkstra(graph, start_node)  
    print(f'Kürzeste Pfade von {start_node}: {shortest_paths}')
```

**Reflexion:**

Das Erlernen und Verstehen von Algorithmen erforderten nicht nur technisches Wissen, sondern auch die Fähigkeit, abstrakte Konzepte verständlich zu kommunizieren. Die Reflexion über verschiedene Algorithmen, ihre Anwendungsfälle und Effizienz trug dazu bei, ein tiefes Verständnis zu entwickeln.

Die Teilnahme an Diskussionen, insbesondere bei komplexen Algorithmen wie Dijkstra, förderte meine analytischen Fähigkeiten und half, verschiedene Herangehensweisen zu verstehen. Das Erklären von Algorithmen gegenüber Peers trug dazu bei, mein Wissen zu festigen und meine Kommunikationsfähigkeiten zu verbessern.

**Zukünftige Schritte:**

Um meine Fähigkeiten weiter zu vertiefen, plane ich, an Projekten teilzunehmen, die den Einsatz fortschrittlicher Algorithmen erfordern. Die kontinuierliche Anwendung und Diskussion dieser Konzepte wird meine Kompetenz in der Algorithmik weiter stärken.