

# Übungsblatt 9 zur Vorlesung Effiziente Algorithmen (Winter 2025/26)

**Abgabe:** Bis 2025-12-19 18:00, on ILIAS.

## 1. Aufgabe

10 + 10 Punkte

Betrachten Sie im Folgenden den (7, 4) Hamming-Code aus der Vorlesung.

- Gegeben sei die Nachricht 0101. Bestimmen Sie die Parity Bits und den zu übertragenden Block.
- Ist 1111111 ein valider Block, d.h. sind (nachweisbare) Fehler aufgetreten?

## 2. Aufgabe

10 + 30 Punkte

Eine einfachere Implementierung der Tiefensuche erhält man über eine rekursive Methode, wie etwa die folgende Java-Implementierung:

```
1 public class DepthFirstSearch {  
2     Graph G; boolean[] visited;  
3  
4     public DepthFirstSearch(Graph G) {  
5         this.G = G; visited = new boolean[G.n];  
6     }  
7  
8     public void dfs(int s) {  
9         visited[s] = true;  
10        for (int v : G.adj[s])  
11            if (!visited[v]) dfs(v);  
12    }  
13 }
```

- a) Welches Problem tritt mit dieser Java-Implementierung schon auf mittelgroßen Graphen auf?
- b) Eine naheliegende Lösung für das Problem aus a) besteht darin, die Rekursion wieder loszuwerden. Die gelingt auch ohne die Iterator-Lösung aus der Vorlesung, etwa mit folgendem, einfachen Java-Code:

---

```
1 public void dfsIterativeSimple(int s) {
2     Stack<Integer> todo = new LinkedStack<>();
3     todo.push(s);
4     while (!todo.empty()) {
5         int v = todo.pop();
6         visited[v] = true;
7         for (int w : G.adj[v])
8             if (!visited[w]) todo.push(w);
9     }
10 }
```

---

Analysieren Sie die Laufzeit und den Speicherplatz von `dfsIterativeSimple` und vergleichen Sie das Ergebnis mit der Tiefensuche aus der Vorlesung.

### 3. Aufgabe

30 Punkte

Gegeben sei ein Digraph  $G$ , sodass es einen Pfad von einem Knoten  $u$  zu einem Knoten  $v$  gibt. Beweisen oder widerlegen Sie: In jeder Tiefensuche wird  $v$  *active* bevor  $u$  *done* wird.

### 4. Aufgabe

20 + 40 Punkte

Ein *Superstar* ist eine Person, die von allen anderen Personen gekannt wird, die selbst aber keine andere Person kennt.

- a) Wenn Sie eine Menge von  $n$  Personen und die Relation „ $a$  kennt  $b$ “ als Digraph modellieren, also „ $a$  kennt  $b$ “  $\iff (a, b) \in E$ , wie können Sie dann einen Superstar charakterisieren?
- b) Sei der Digraph aus Teilaufgabe a) als  $n \times n$  Adjazenzmatrix  $A$  gegeben. Entwerfen Sie einen Algorithmus, der bei Eingabe  $A$  in Zeit  $o(n^2)$  entscheidet, ob die durch  $A$  modellierte Population einen Superstar besitzt.

Begründen Sie die Korrektheit Ihres Algorithmus und weisen Sie eine dem Zwecke der Aufgabenstellung dienliche Laufzeitschranke nach.