HF-ICT - Höhere Fachschule für Informations- und Kommunikationstechnologie
Programmieren 3. Semester, Algorithmen und Datenstrukturen
David Herzig

Algorithmen und Datenstrukturen

Exercise sheet 6

---

1 Graph - All nodes are reachable

Gegeben ist ein gerichteter Graph in der Form wie im Unterricht besprochen:

```
Graph g;
```

Implementieren Sie die folgende Funktion

```
bool allNodeAreReachable(Graph *g,
                         int start);
```

Diese Funktion erhält als Parameter einen Graphen, die Anzahl Knoten und den Start Knoten. Die Funktion liefert `true` falls alle anderen Knoten im Graphen von `start` erreichbar sind. Falls ein oder mehrere Knoten nicht erreichbar sind, so liefert die Funktion `false`.

2 Graph - Shortest Reach

Gegeben ist ein ungerichteter Graph in der Form wie im Unterricht besprochen:

```
Graph g;
```

Implementieren Sie die folgende Funktion

```
int shortestReach(Graph *g,
                  int start, int end);
```

Diese Funktion liefert die minimale Anzahl Knoten, welche von `start` nach `end` passiert werden muss. Falls `start` und `end` gleich sind, liefert die Funktion `0`. Falls kein Weg zwischen `start` und `end` existiert, so liefert die Funktion den Wert `-1`.

3 Graph - Get Path

Gegeben ist ein gerichteter, gewichteter Graph in der Form wie im Unterricht besprochen:

```
Graph g;
```

Implementieren Sie die folgende Funktion

```
vector<int> getPath(Graph *g
                    int start, int end);
```

Diese Funktion liefert einen Vector, welche alle Knoten (günstigster Weg) beinhaltet, welche von `start` nach `end` passiert werden müssen.