

## Digraph-Datenstruktur (9 Punkte)

Vervollständigen Sie das auf der Kurs-Homepage verfügbare Skelett für `VertexPropertyMap` und implementieren Sie eine Unterklasse `DirectGraph` von `DirectedGraphBase`, die die abstrakten Methoden implementiert.

`void dfs(Node startNode, GraphVisitor visitor)` und `void bfs(Node startNode, GraphVisitor visitor)` sollen dabei so implementiert werden, dass dabei die entsprechenden Visitor-Methoden aufgerufen werden. Da diese Methoden unabhängig vom Visitor funktionieren müssen, ist es erforderlich, ein `BitSet` zur Verwaltung der bereits besuchten Knoten mitzuführen.

Testcode, um ihre Datenstruktur zu überprüfen, ist ebenfalls vorhanden.

## Graphalgorithmen mittels Visitor

### Topologische Ordnung (3 Punkte)

Implementieren Sie den in der Vorlesung vorgestellten Visitor zur Berechnung einer topologischen Ordnung der Knoten.

### Discovery- und Finishing-Timestamps (3 Punkte)

Implementieren Sie einen Visitor (`CollectTimesVisitor`), der für DFS den jeweiligen discovery- und finishing-Zeitpunkt in einem property-Vektor (`VertexPropertyMap`) verzeichnet. Dieser soll nach dem Durchlauf der DFS aus dem Visitor ausgelesen werden können.

---

Abgabetermin ist der 10. Januar

Bitte nur in gewohnter Form an [steffen@dfki.de](mailto:steffen@dfki.de)