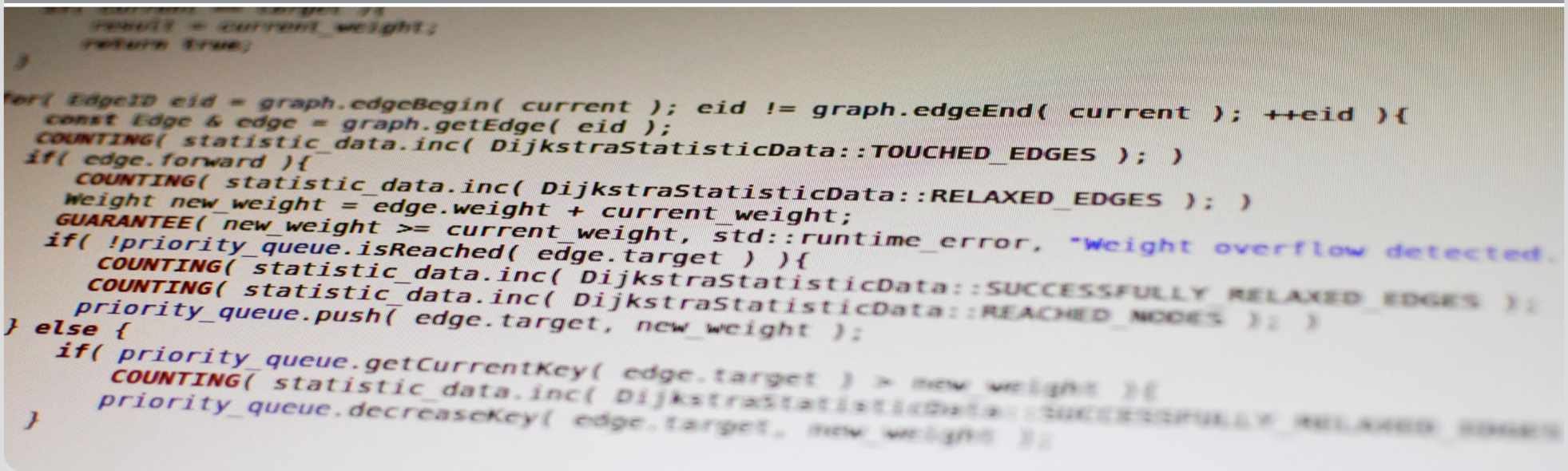
Karlsruher Institut für Technologie

Übung 3 – Algorithmen II

Moritz Laupichler, Nikolai Maas – *{moritz.laupichler, nikolai.maas}@kit.edu*

`https://algo2.iti.kit.edu/AlgorithmenII_WS23.php`

Institut für Theoretische Informatik - Algorithm Engineering



Themenübersicht

- SCCs mit DFS berechnen
- Besprechung Übungsblatt 1

Starke Zusammenhangskomponenten

SCC

Invariante 1

Keine Kanten von geschlossenen in offene Komponenten.

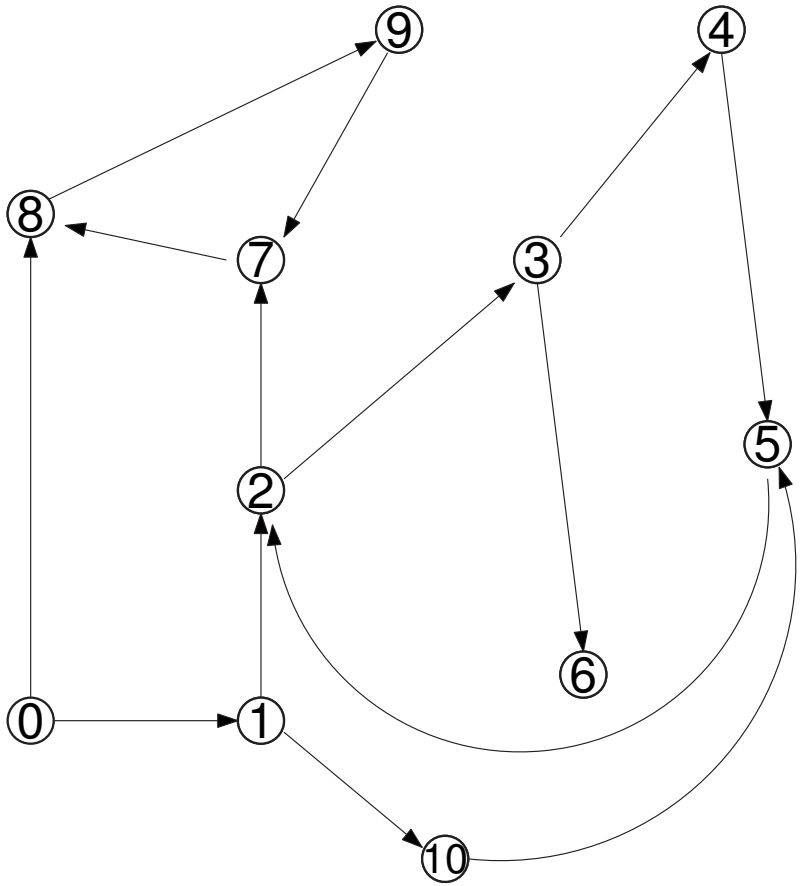
Invariante 2

Offene Komponenten liegen auf Pfad.

Invariante 3

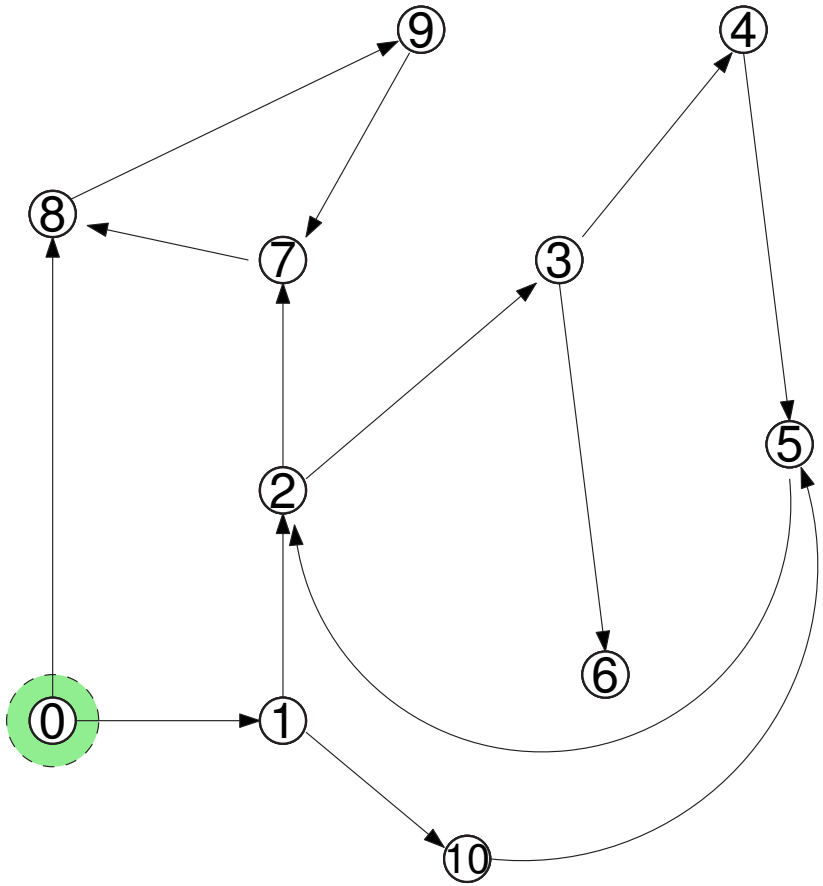
Repräsentanten partitionieren offene Komponenten bzgl. dfsNum.

SCC



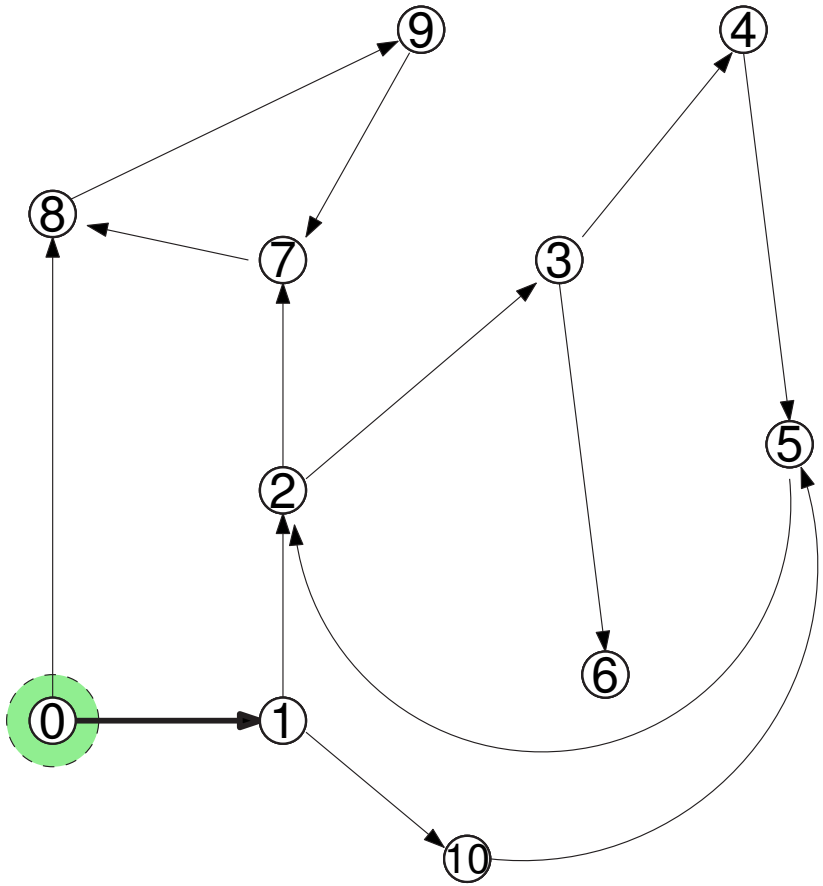
oNodes	oReps

SCC



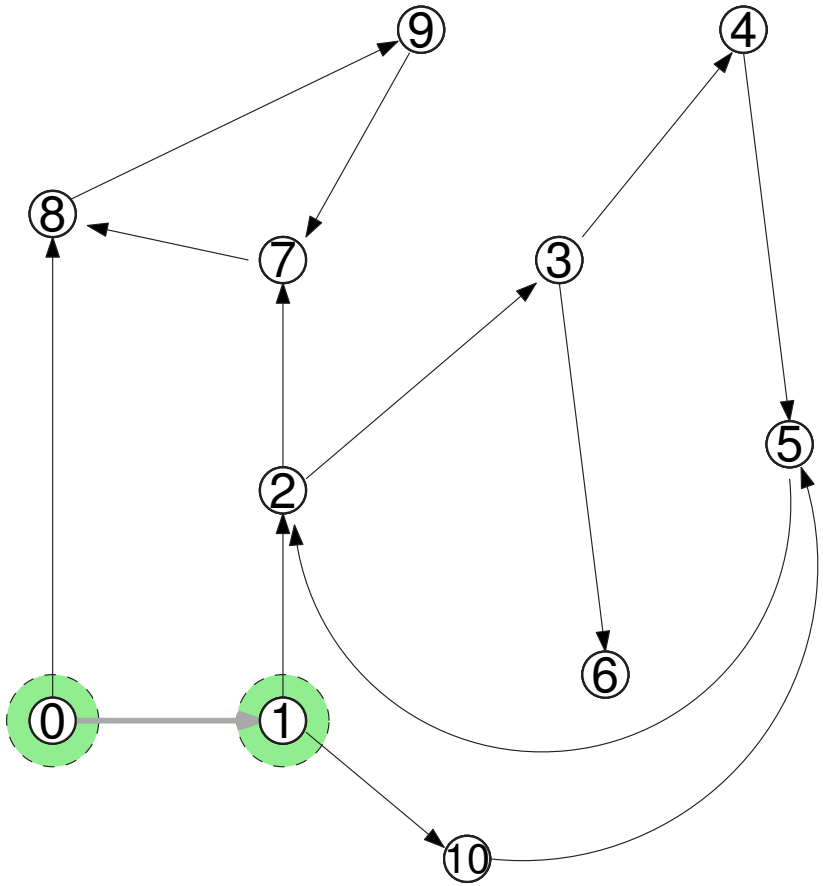
oNodes	oReps
0	0

SCC



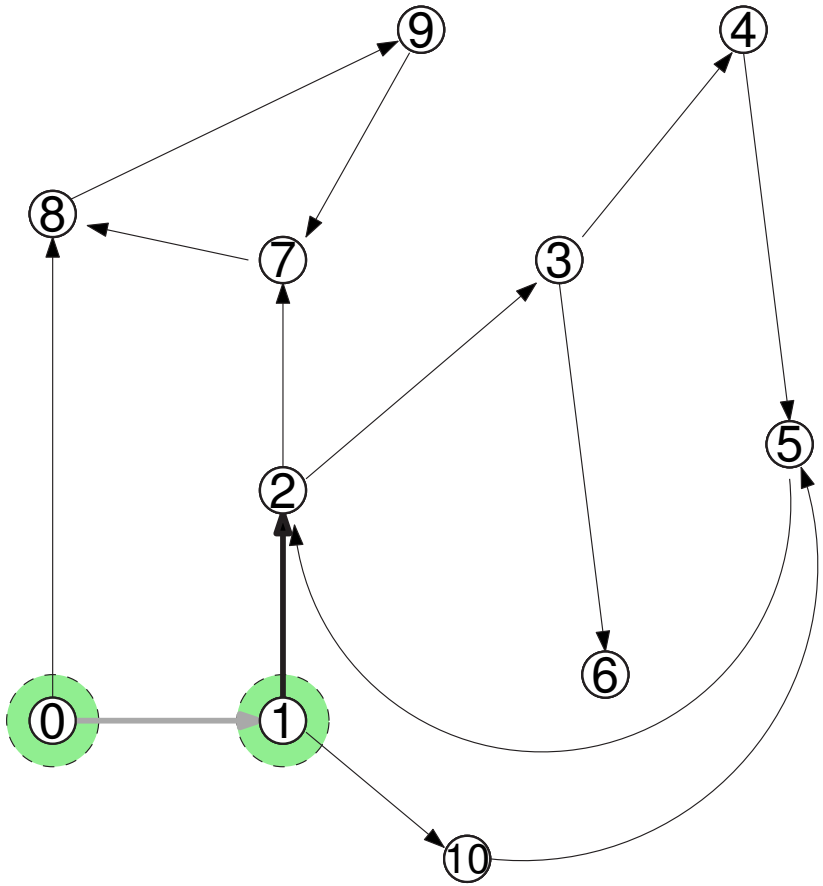
oNodes	oReps
0	0

SCC



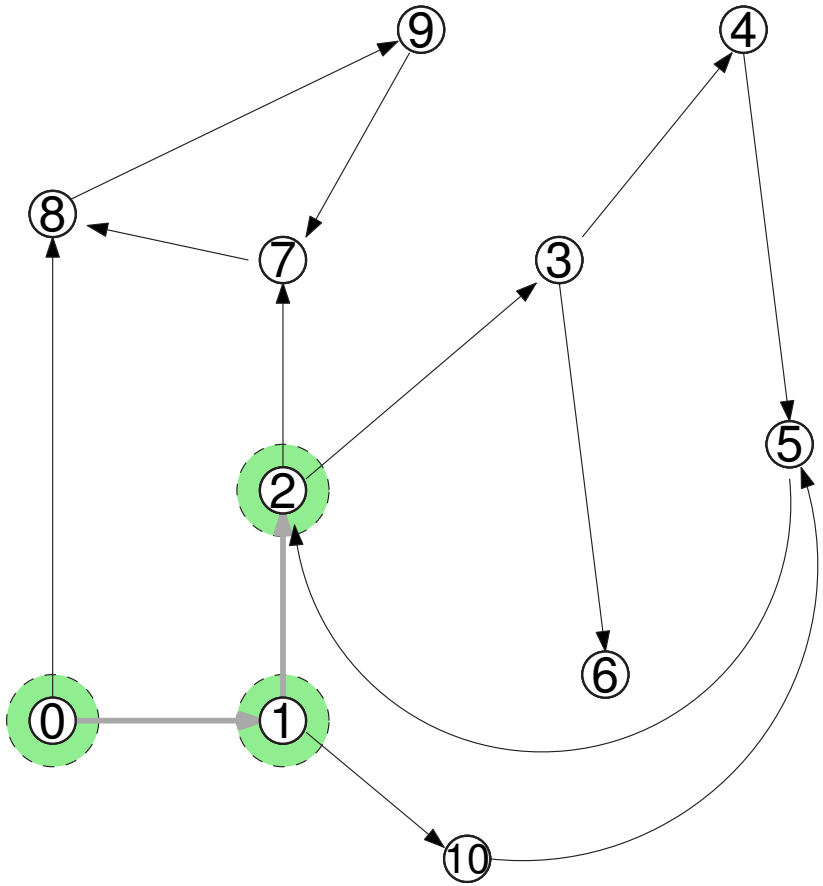
oNodes	oReps
0	0
1	1

SCC



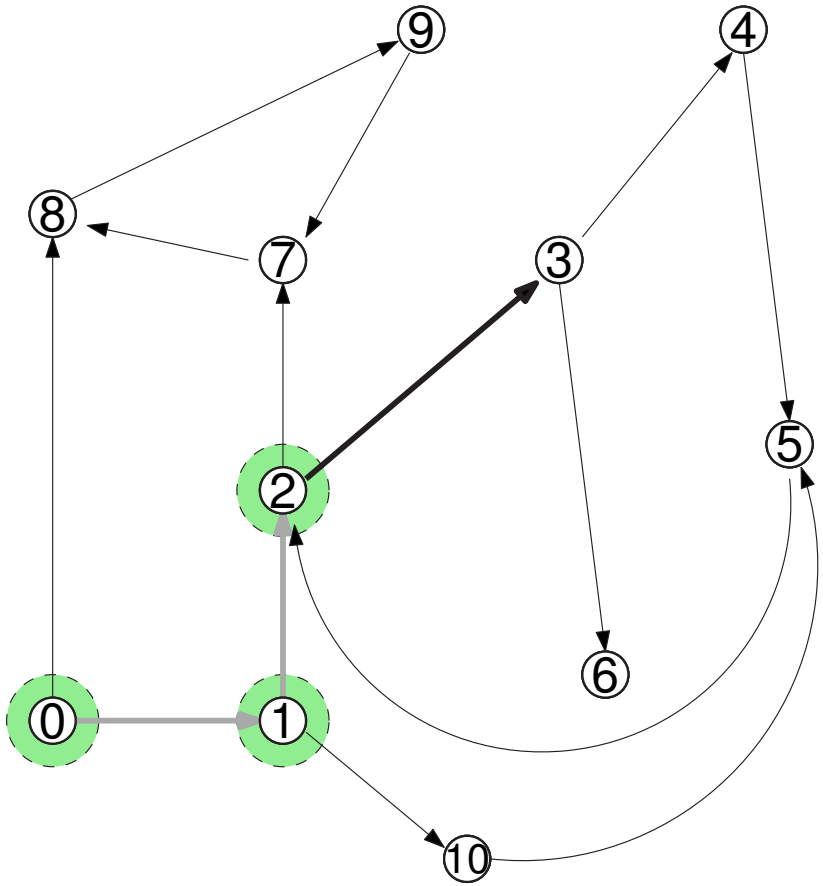
oNodes	oReps
0	0
1	1

SCC



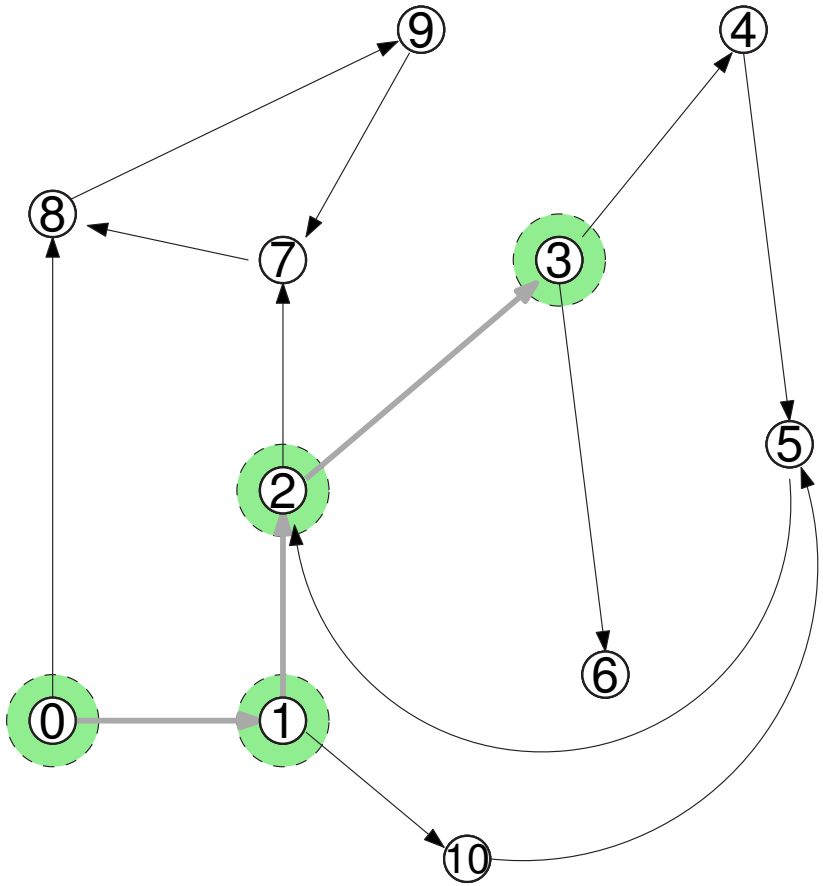
oNodes	oReps
0	0
1	1
2	2

SCC



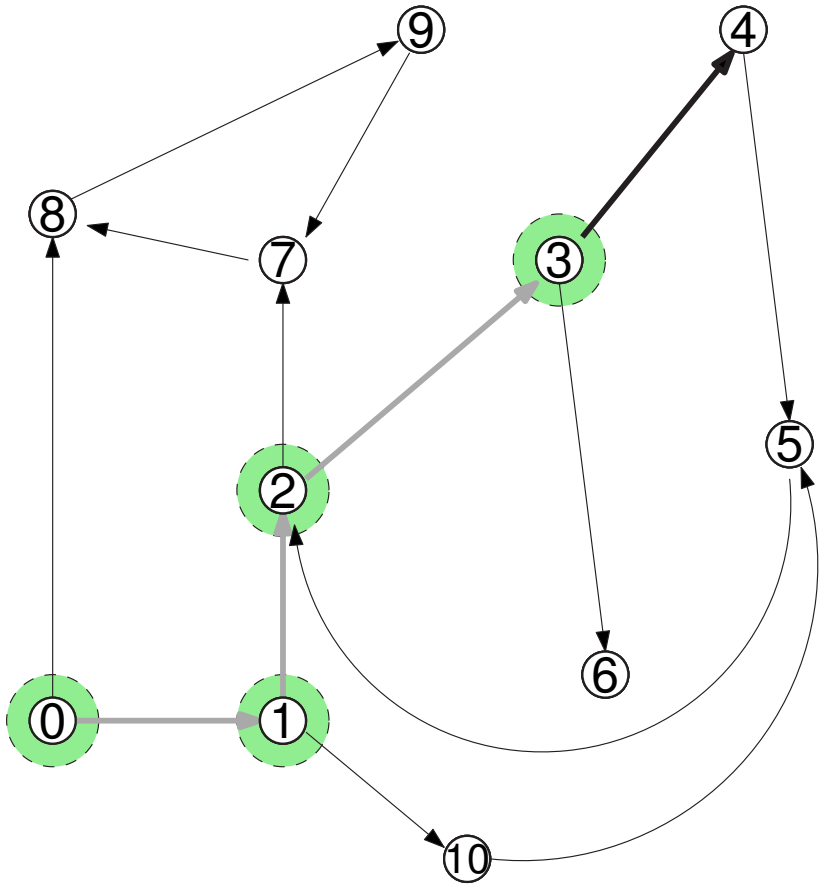
oNodes	oReps
0	0
1	1
2	2

SCC



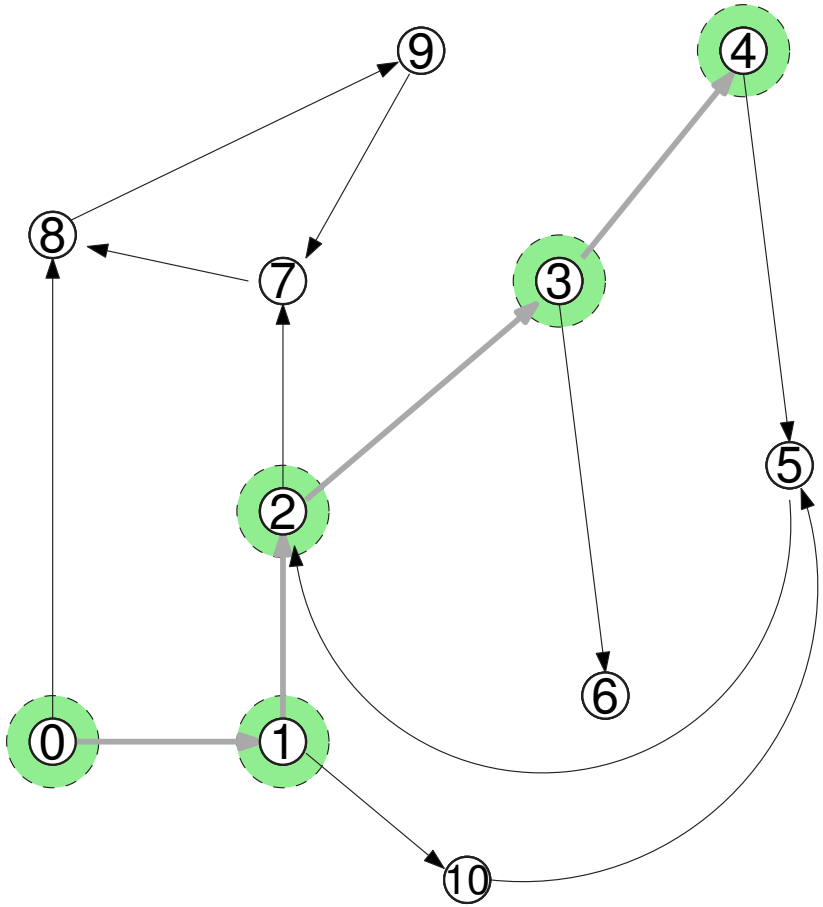
oNodes	oReps
0	0
1	1
2	2
3	3

SCC



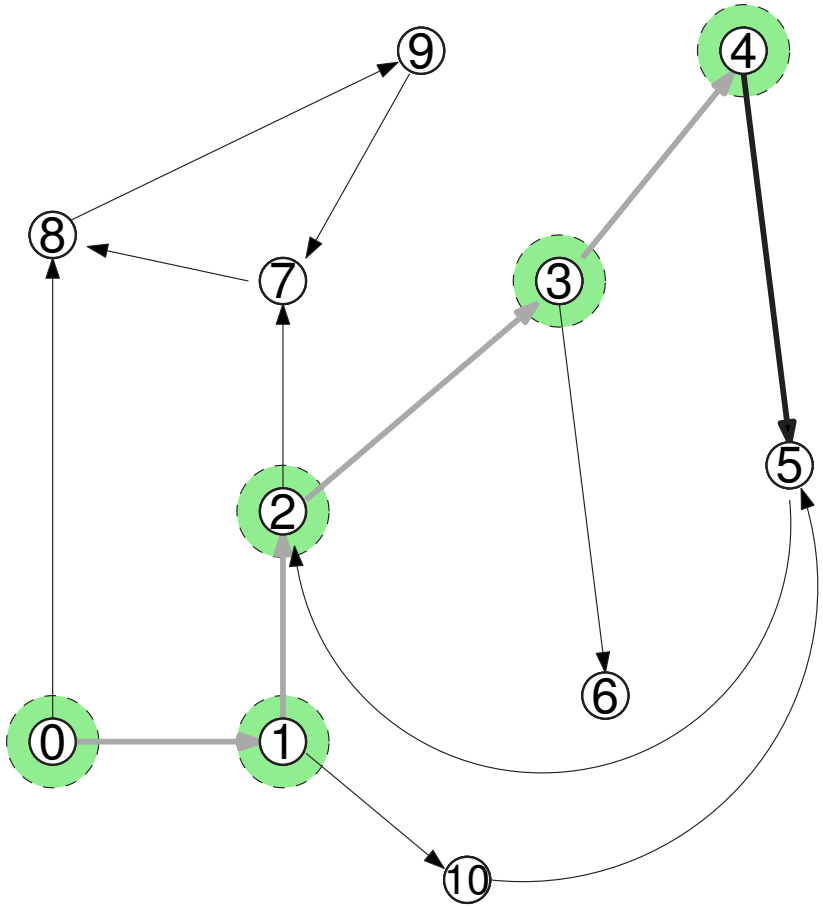
oNodes	oReps
0	0
1	1
2	2
3	3

SCC



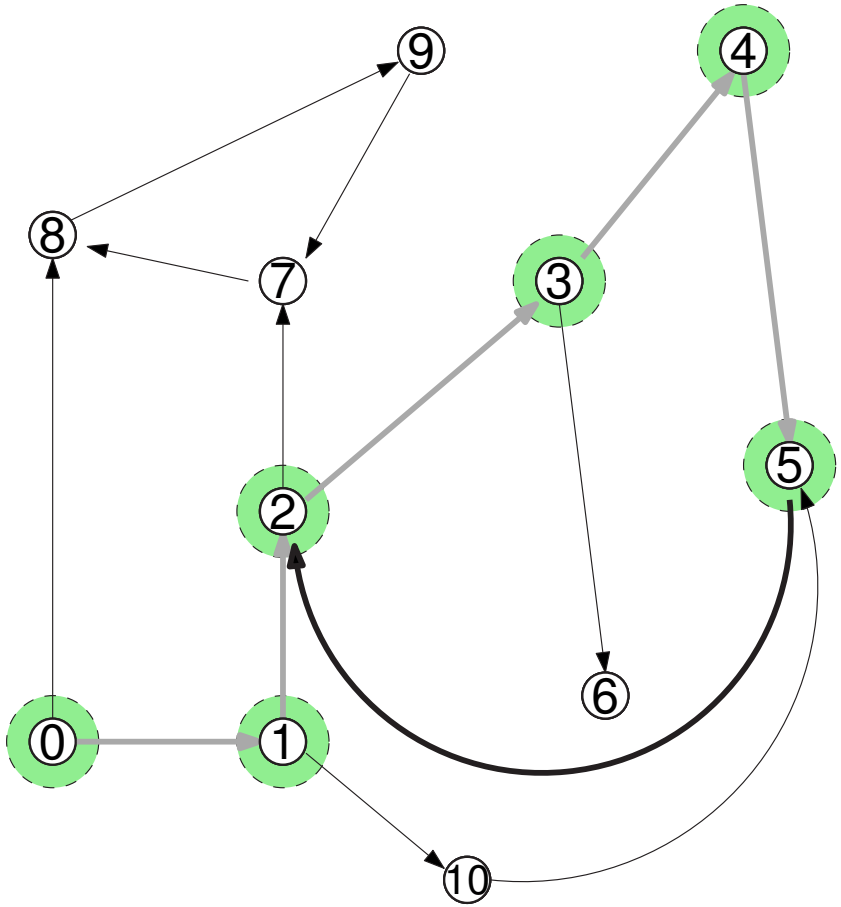
oNodes	oReps
0	0
1	1
2	2
3	3
4	4

SCC



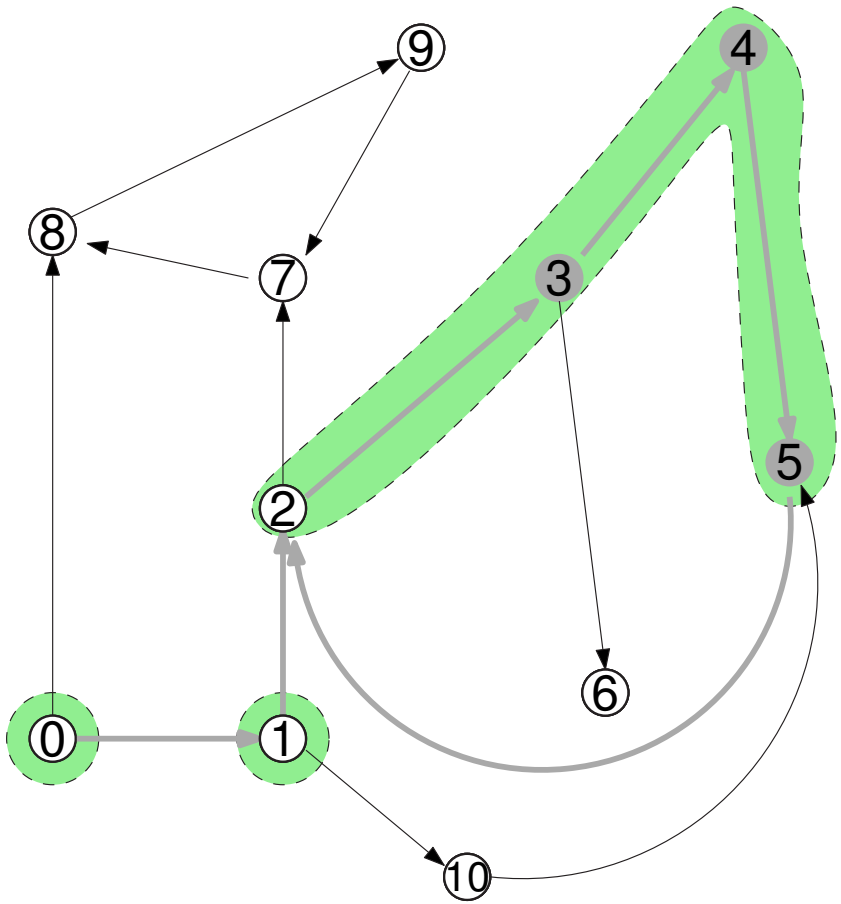
oNodes	oReps
0	0
1	1
2	2
3	3
4	4

SCC



oNodes	oReps
0	0
1	1
2	2
3	3
4	4
5	5

SCC



oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC

Invariante 1

Keine Kanten von geschlossenen in offene Komponenten.

- Tiefensuche sucht Pfad durch Graphen
- Nur Rückwärtskanten ergeben Kreise
- Kreise vereinigen alle auf dem Kreis liegenden offenen Komponenten

Invariante 2

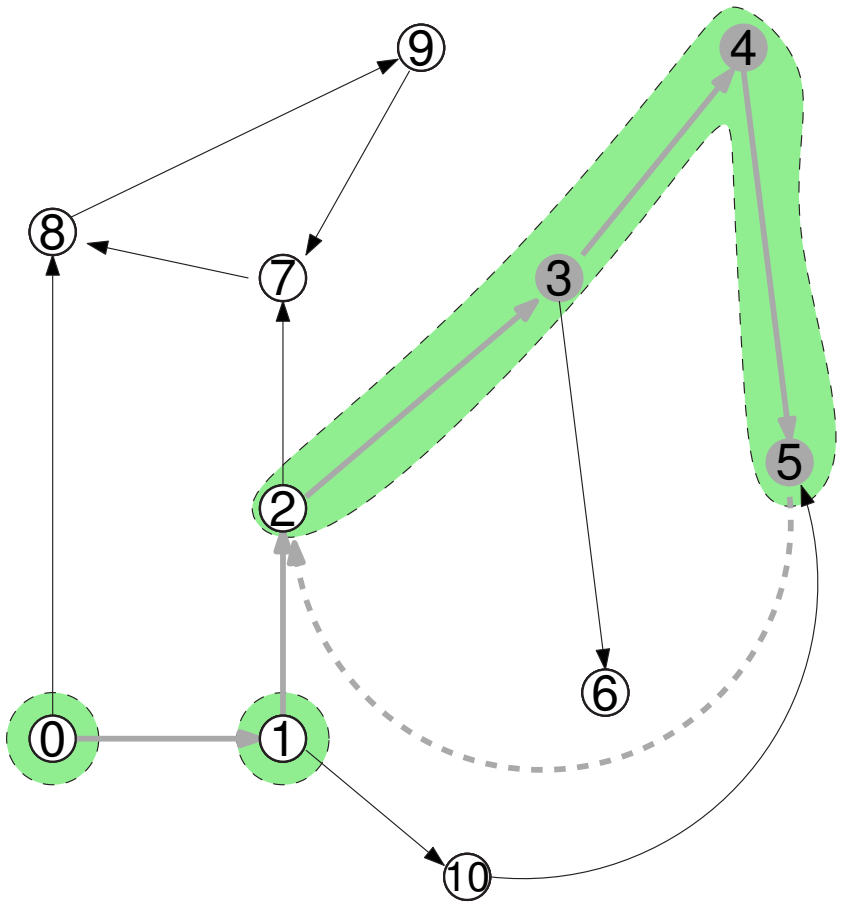
Offene Komponenten liegen auf Pfad.

- Repräsentant ist minimal auf Kreis

Invariante 3

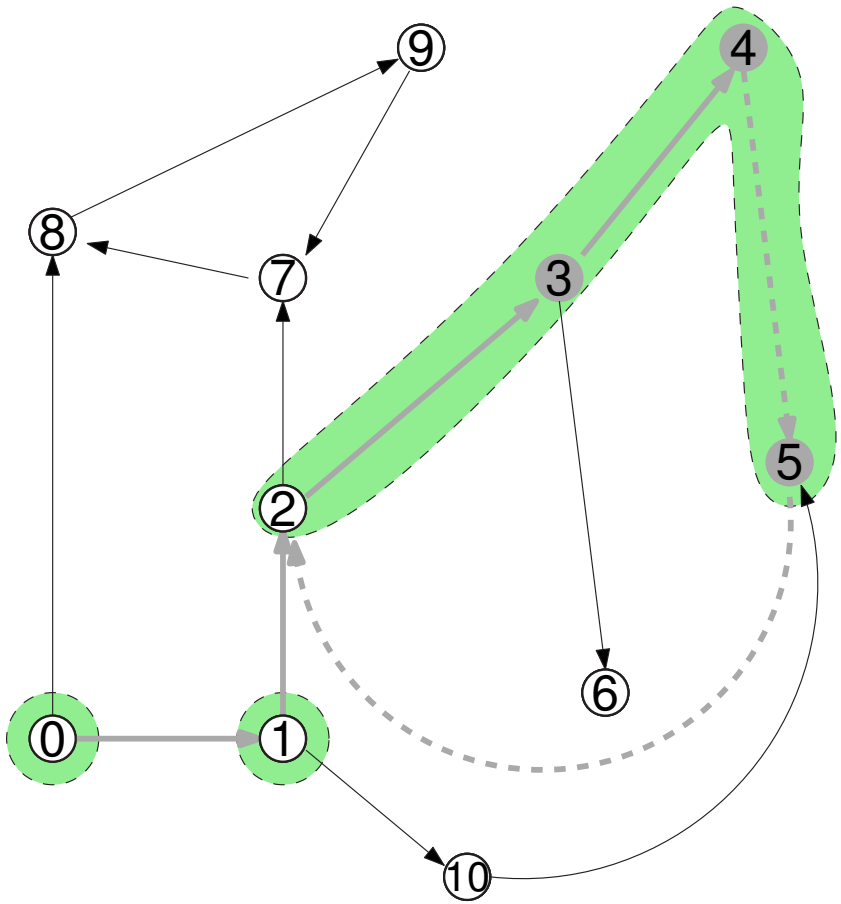
Repräsentanten partitionieren offene Komponenten bzgl. dfsNum.

SCC



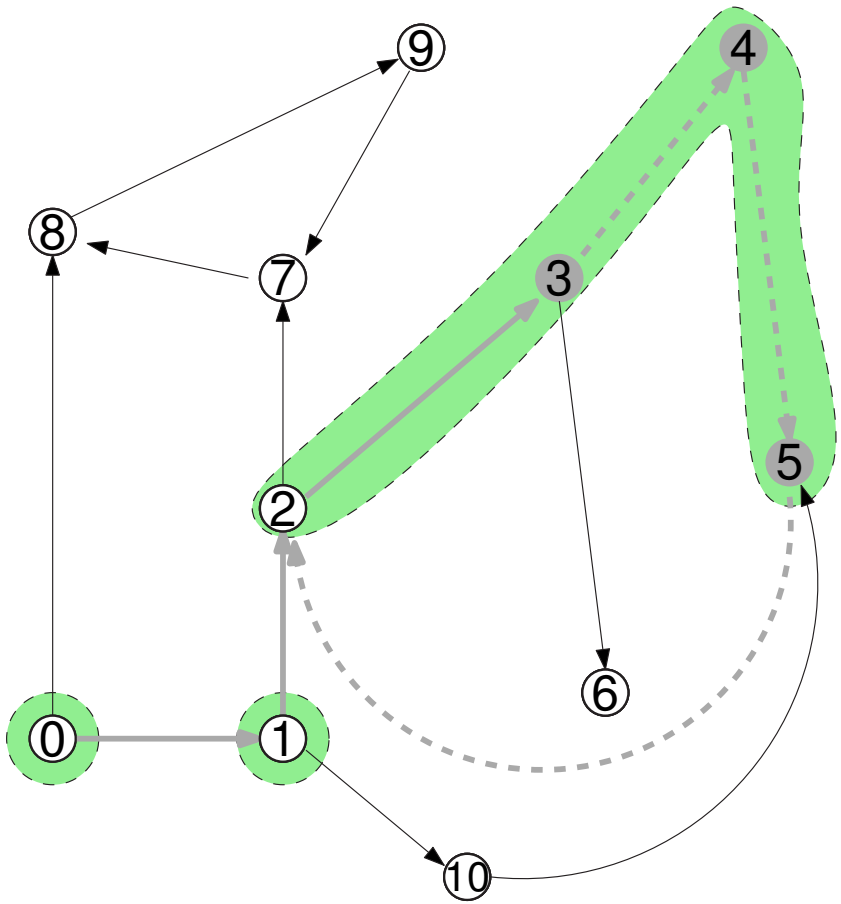
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



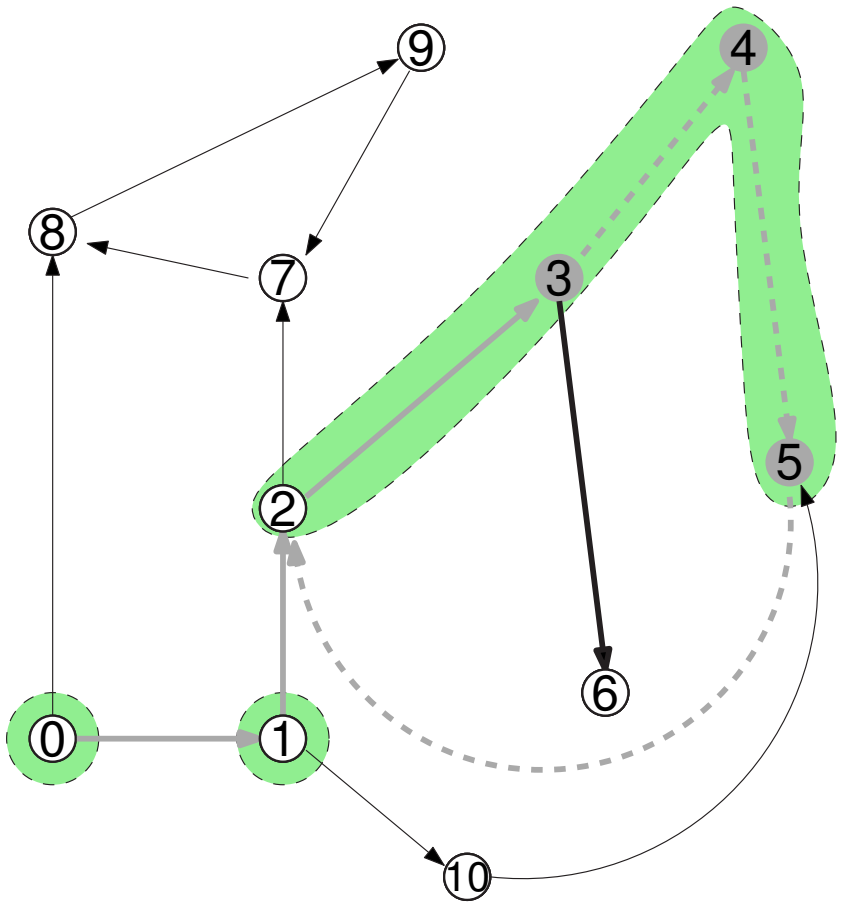
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



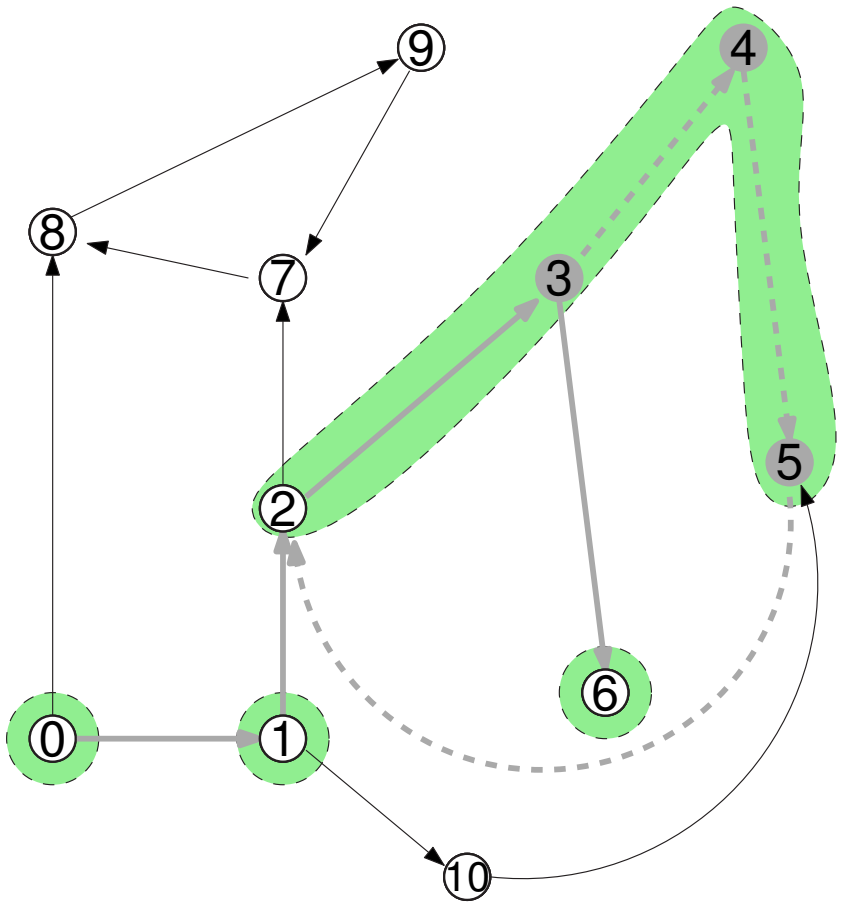
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



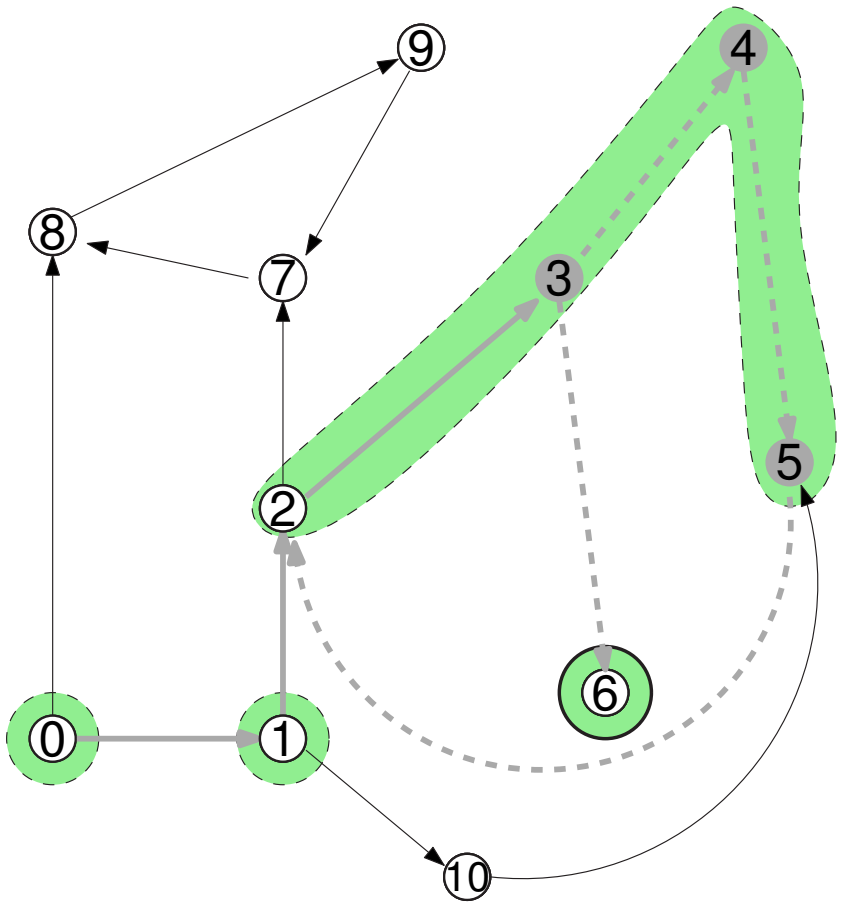
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
6	6

SCC



oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
6	6

SCC

- Komponenten werden nach Bearbeitung aller ausgehenden Kanten geschlossen
- Alle offenen Komponenten liegen auf Stack
- Kante von geschlossener in offene Komponenten hätte bei Bearbeitung Kreis induziert

Invariante 1

Keine Kanten von geschlossenen in offene Komponenten.

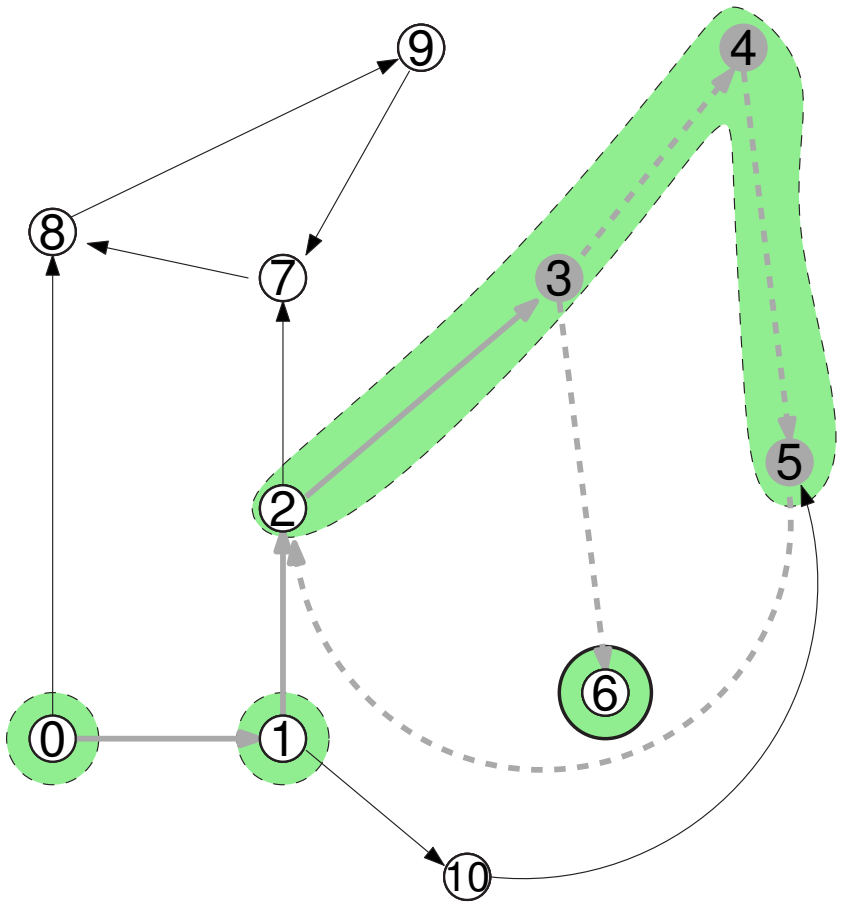
Invariante 2

Offene Komponenten liegen auf Pfad.

Invariante 3

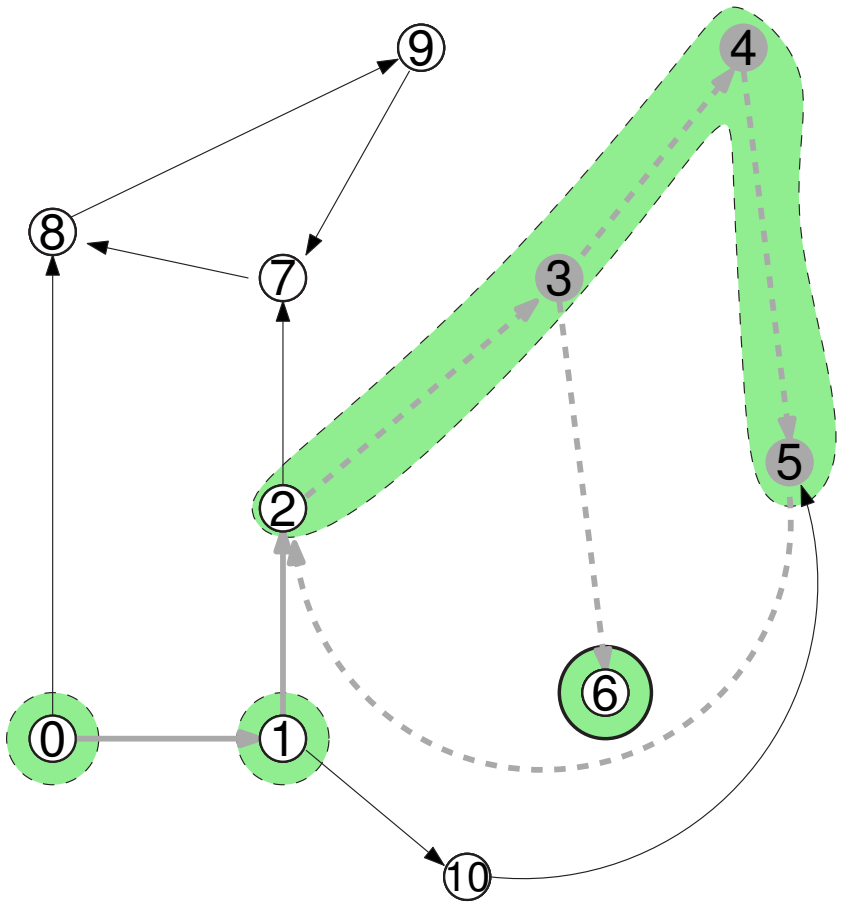
Repräsentanten partitionieren offene Komponenten bzgl. dfsNum.

SCC



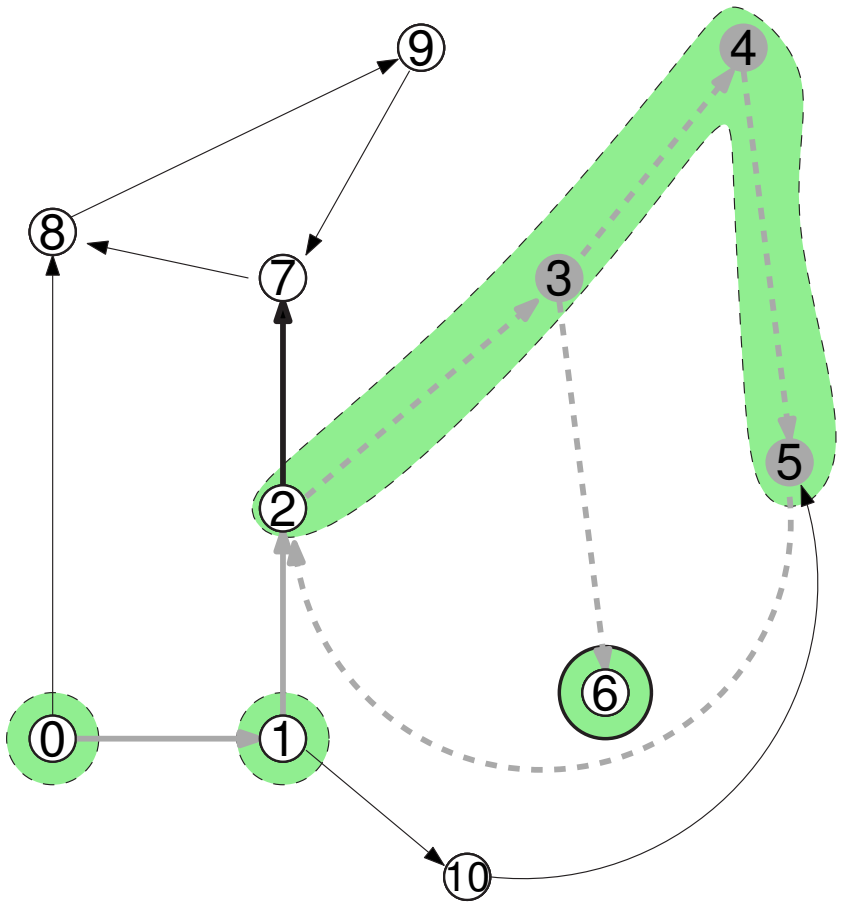
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



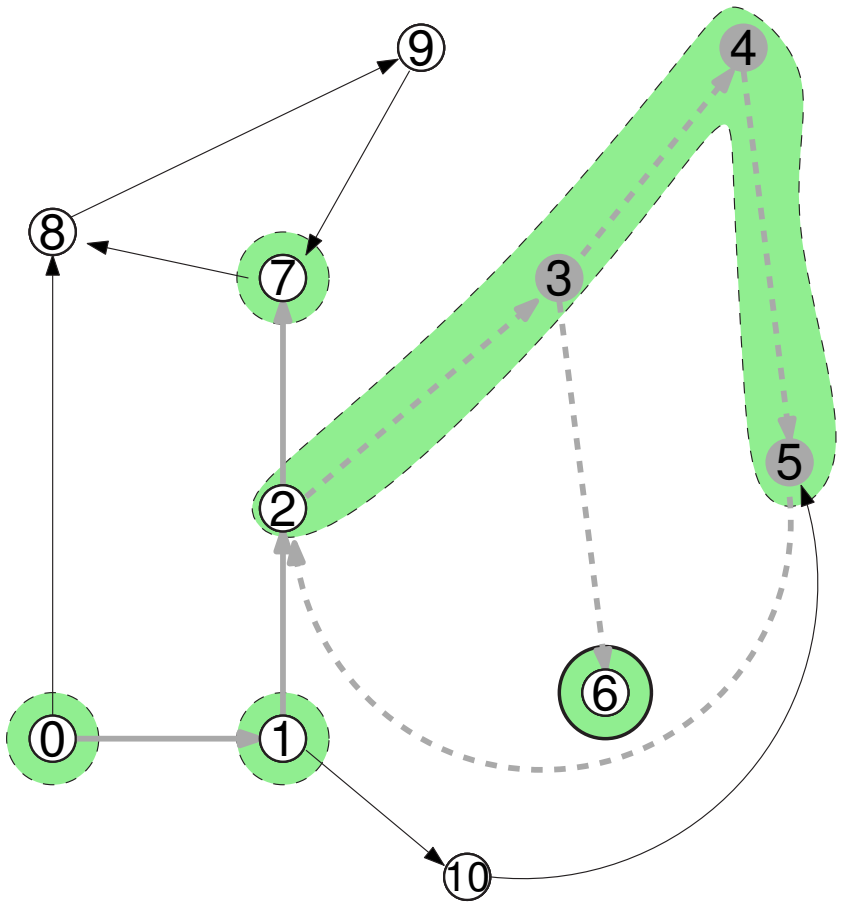
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



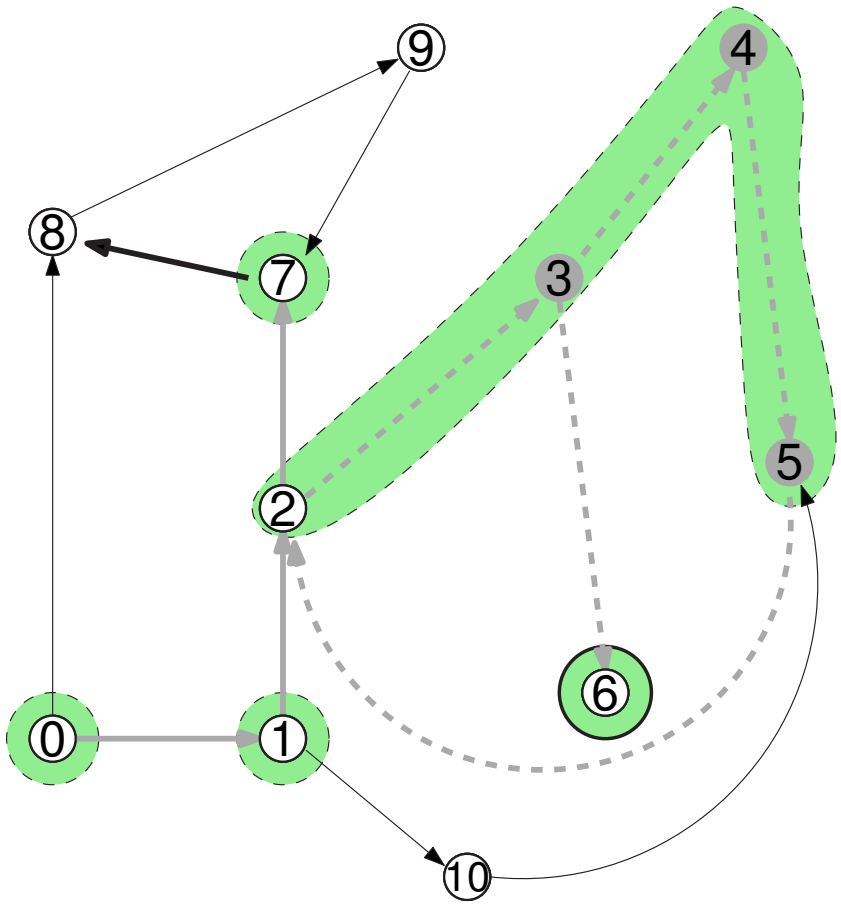
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

SCC



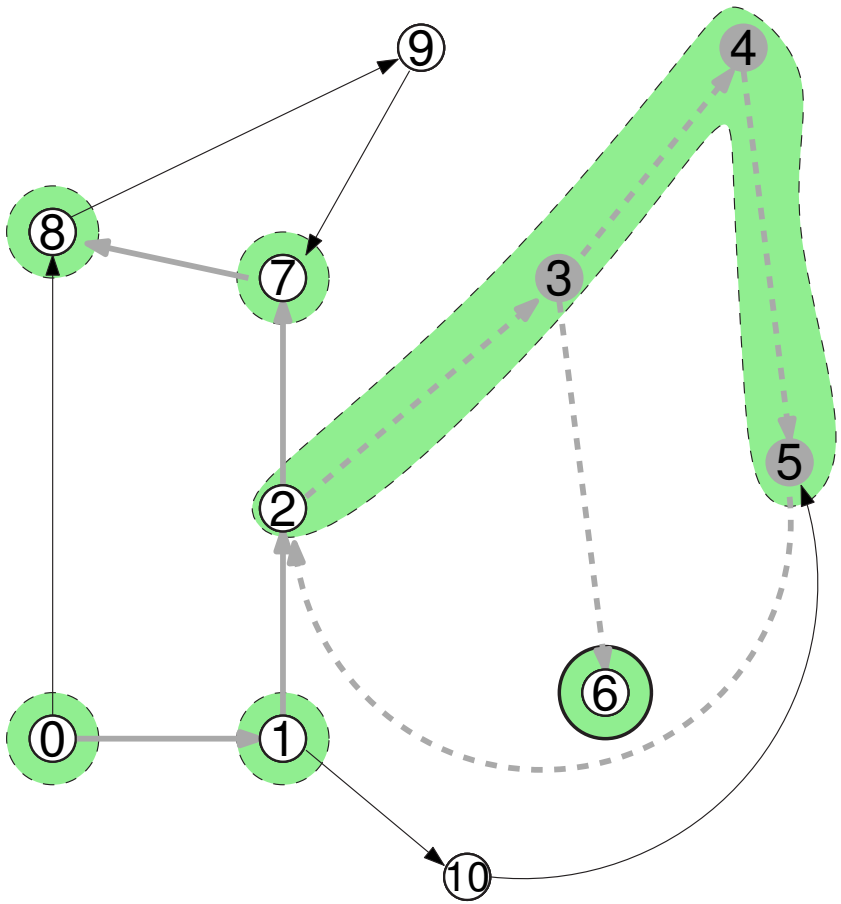
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7

SCC



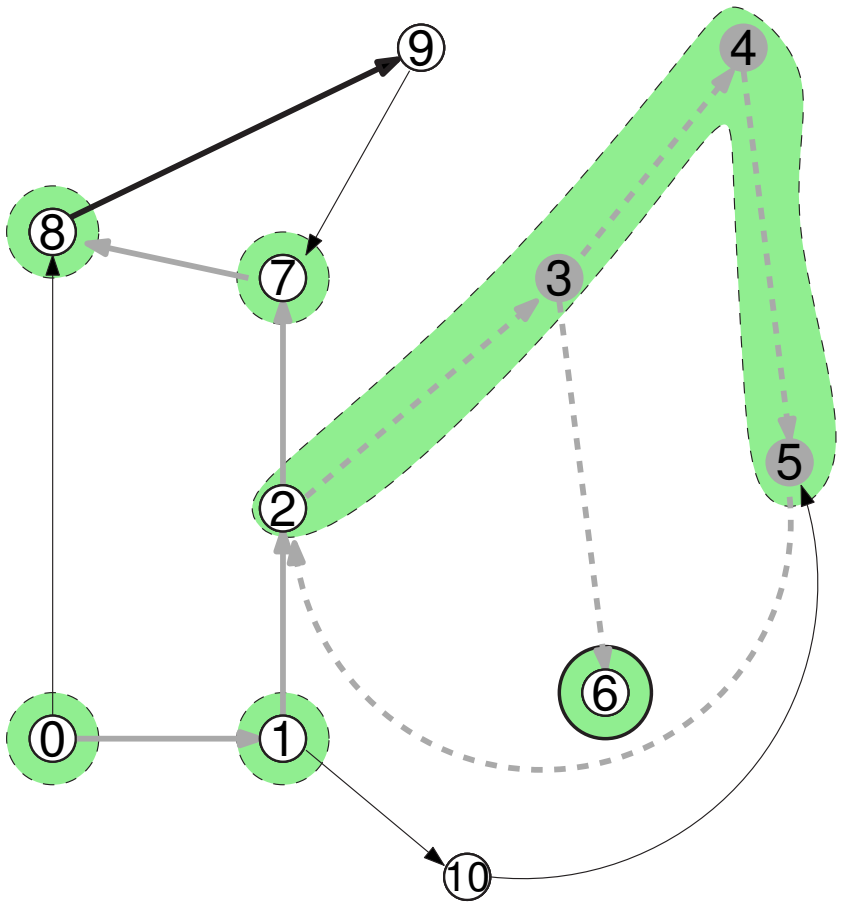
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7

SCC



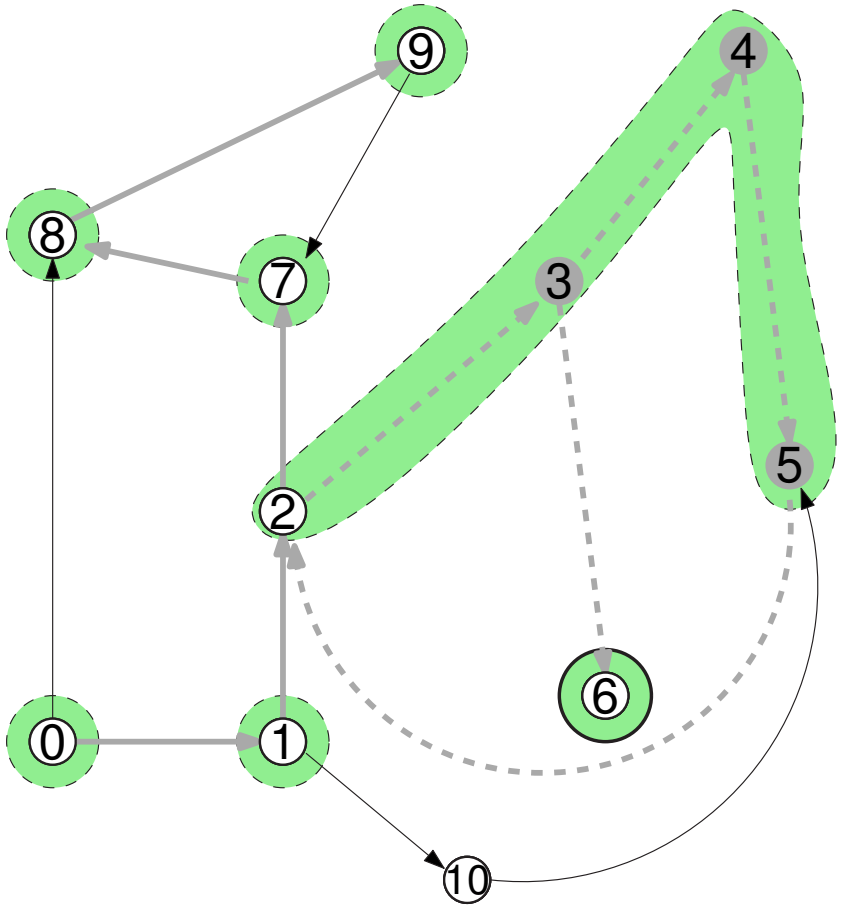
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	8

SCC



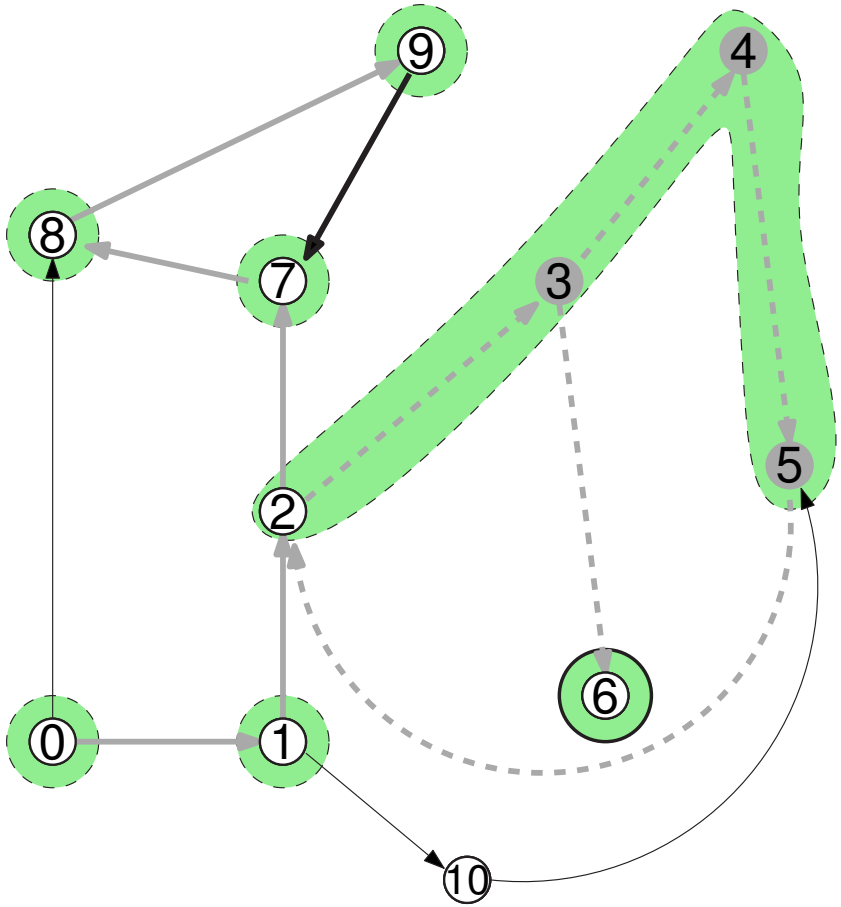
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	8

SCC



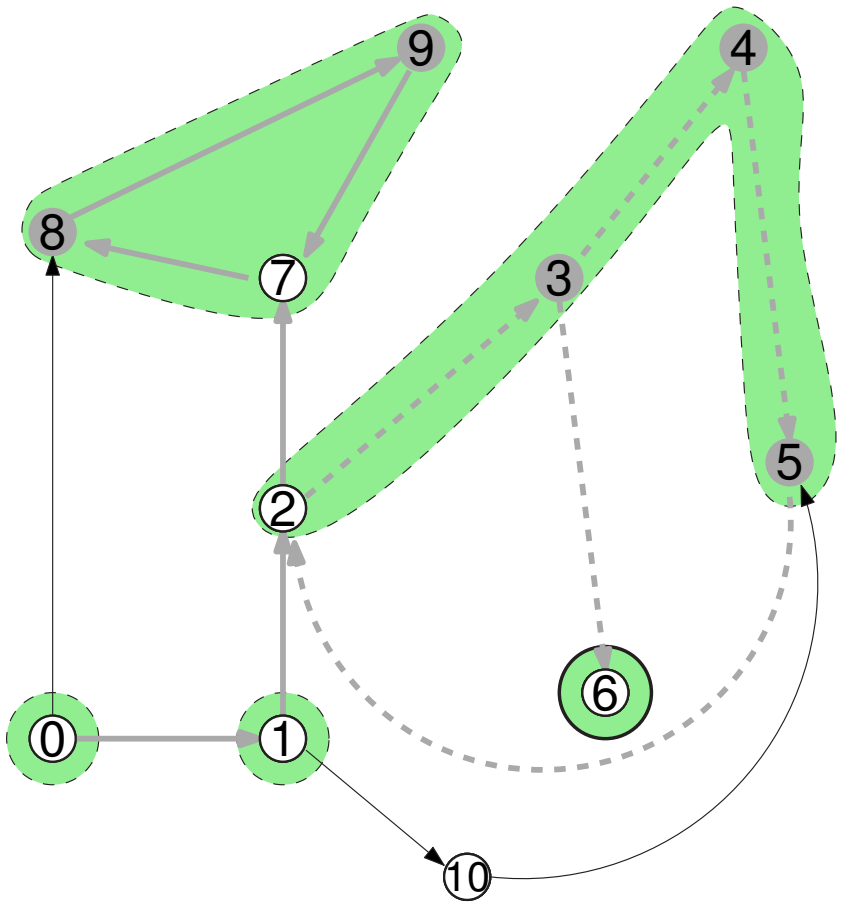
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	8
9	9

SCC



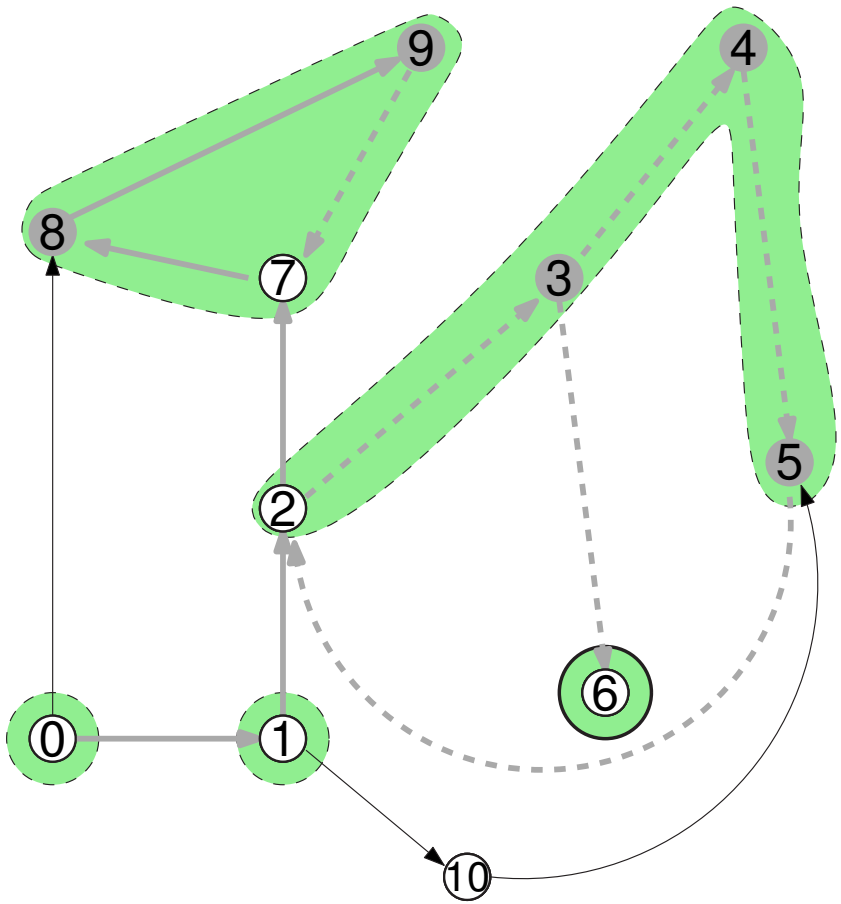
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	8
9	9

SCC



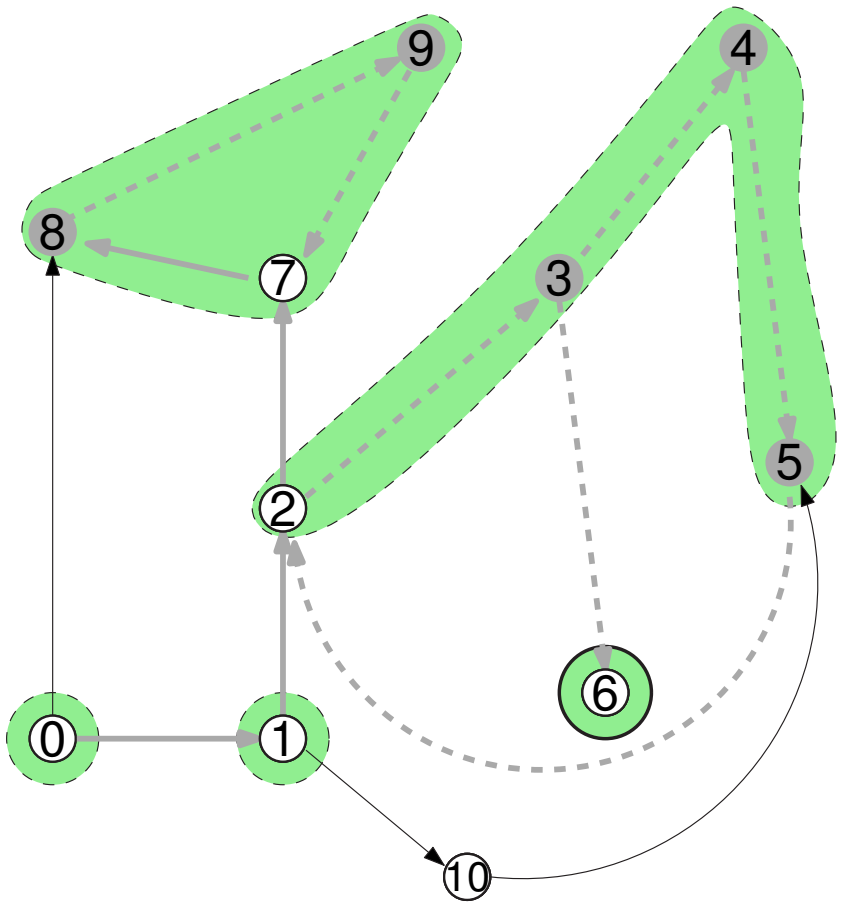
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	
9	

SCC



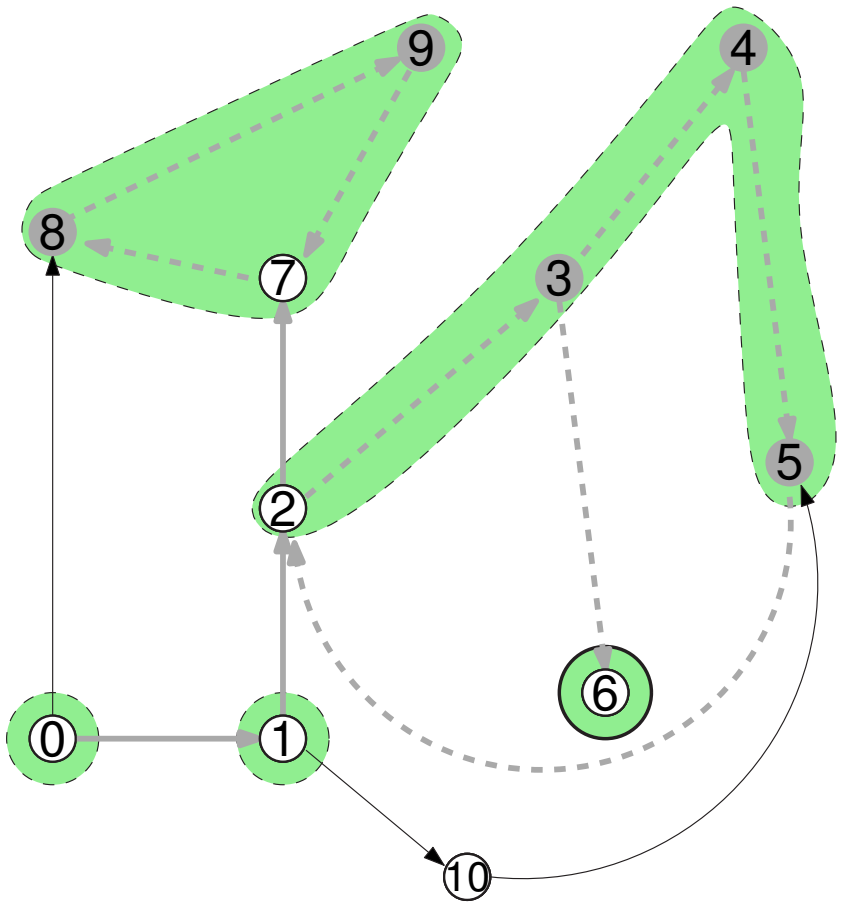
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	
9	

SCC



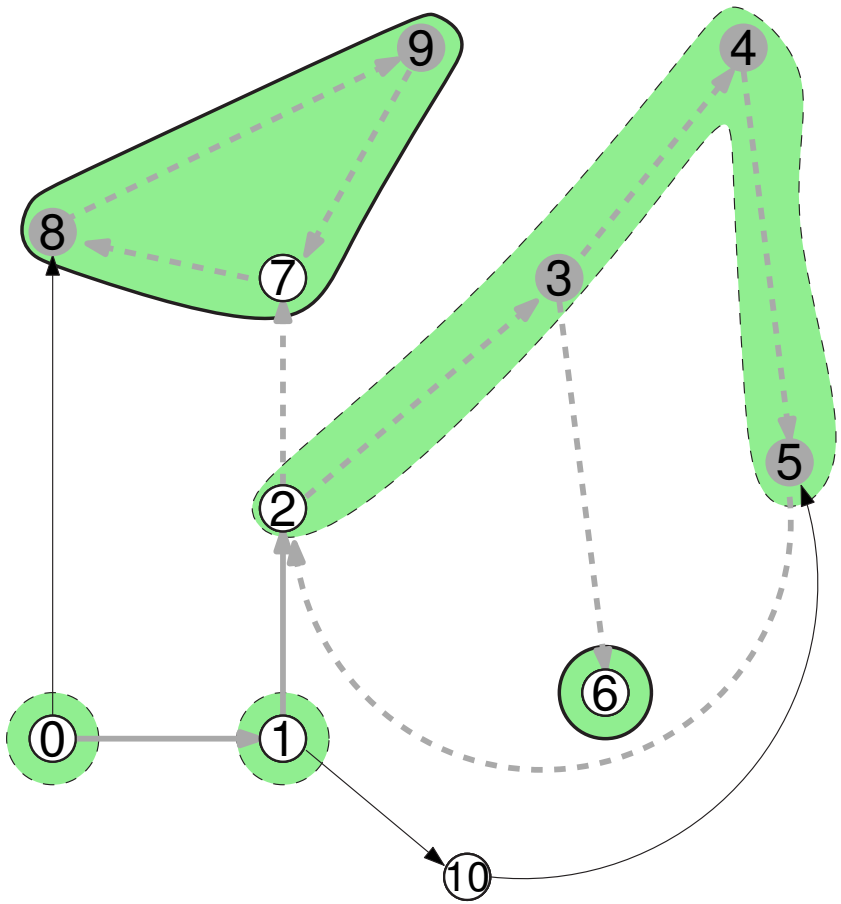
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	
9	

SCC



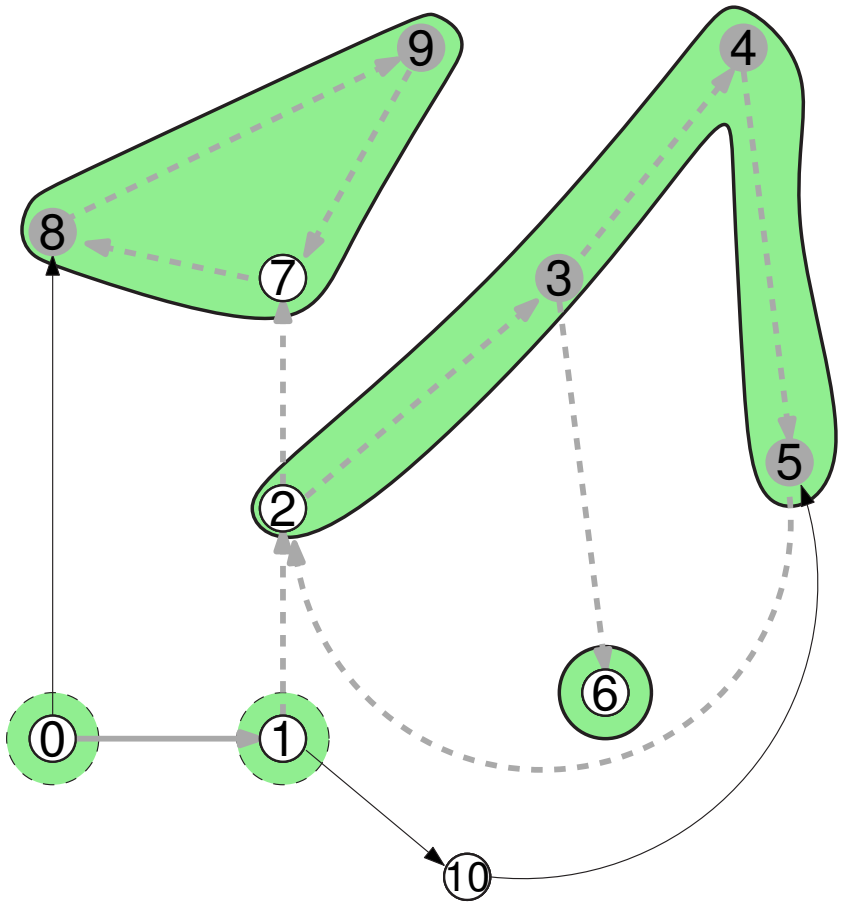
oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	
9	

SCC



oNodes	oReps
0	0
1	1
2	2
3	
4	
5	
7	7
8	
9	

SCC



oNodes	oReps
0	0
1	1
2	2
3	
4	
5	

Besprechung Übungsblatt 1

- Aufgabe 3: Best-Case Verhalten von Fibonacci-Heaps
- Aufgabe 7: Laufzeit von Dijkstra

Ende!

Feierabend!