

Übungsblatt 9

Übung 9.1 Das bereitgestellte Script *readingorder.py* erzeugt den gerichteten azyklischen Graphen zur Ermittlung der Lesereihenfolge. Der Graph benutzt die Gamera Graph-API, von der Sie die folgenden Methoden benötigen:

```
for node in G.get_nodes(): # Iteration über alle Knoten
    # Iteration über alle von 'node' erreichbaren Knoten
    for nextnode in node.nodes:
```

Bestimmen Sie aus dem Graphen die Lesereihenfolge der Textzeilen und zeichnen Sie Verbindungslinien (*draw_line*) in Lesereihenfolge zwischen den linken Randpunkten der Textzeilen. Der Algorithmus zur topologischen Sortierung eines gerichteten Graphen G lautet:

```
1:  $S \leftarrow []$  ▷ Liste der sortierten Segmente
2: procedure DFS_VISIT(Knoten:  $n$ )
3:    $n.data.label \leftarrow 1$  ▷ markiere Knoten als besucht
4:   for  $m$  in Nachbarknoten( $n$ ) do
5:     if  $m.data.label = 0$  then ▷  $m$  noch nicht abgearbeitet
6:       DFS_VISIT( $m$ )
7:     end if
8:   end for
9:    $S.append(n.data.segment)$  ▷ append fügt abgearbeitetes  $n$  hinten an
10: end procedure
11: for  $n$  in  $G.nodes$  do
12:   if  $n.data.label = 0$  then ▷  $n$  noch nicht abgearbeitet
13:     DFS_VISIT( $n$ )
14:   end if
15: end for
16:  $S.reverse()$  ▷ wegen Verwendung von append ist  $S$  falsch rum geordnet
```

Übung 9.2 Die folgende Abbildung zeigt eine Ground-Truth Segmentierung G und eine von einem Algorithmus erzeugte Segmentierung S . Welche Fehlerrate ergibt sich für die Segmentierung S nach der Methode von Thulke et al.?

