

Topologisches Sortieren (einfache Version)

```
void topsort() throws CycleFound {
    Vertex v, w;

    for (int i = 0; i < vertices.size(); i++) {
        v = findVertexOfIndegreeZero();
        if (v == null) {
            throw new CycleFound();
        }

        // process v, e.g. print v to console

        for each w adjacent to v {
            w.indegree--
        }
    }
}
```

Topologisches Sortieren (effiziente Version)

```
void topsort() throws CycleFound {
    Queue q = new Queue(); // e.g. use LinkedList
    int counter = 0;
    Vertex v, w;

    for each vertex v in graph
        if (v.indegree == 0)
            q.enqueue(v);

    while (!q.isEmpty()) {
        v = q.dequeue();
        // process v, e.g. print it to console
        counter++;

        for each w adjacent to v
            if (--w.indegree == 0)
                q.enqueue(w);
    }

    if (counter != vertices.size())
        throw new CycleFound();
}
```