

```
// Operationen zum Setzen und lesen von Knotendaten
```

```
void G.set_value(node v,int x)
int G.get_value(node v)
```

## 2.1 Topologisches Sortieren

Knotendaten: temporäre indeg-Werte

```
list<node> TopSort(const graph& G)
{
    // berechnet Liste der Knoten von G in topologischer Reihenfolge

    list<node> result;
    stack<node> ZERO;

    // Initialisierung
    node v;
    forall_nodes(v,G) {
        int indeg = G.indeg(v);
        G.set_value(v,indeg);
        if (indeg == 0) ZERO.append(v);
    }

    // Hauptschleife
    while (!ZERO.empty())
    { node v = ZERO.pop();
      result.append(v);
      edge e;
      forall_out_edges(e,v) {
          node w = G.target(e);
          int indeg_w = G.get_value(w);
          if (--indeg_w == 0) ZERO.push(w);
          G.set_value(w,indeg_w);
      }
    }

    return result;
}
```