

Methode `erreichbarIn` mit einem Breitendurchlauf

```

public List<Vertex> erreichbarIn(String pStadt, double pMaxEntfernung){
    Vertex pVertex = karte.getVertex(pStadt);
    List<Vertex> ergebnis = new ListWithViewer<Vertex>();
    List<Double> entfernungen = new ListWithViewer<Double>();
    ergebnis.append(pVertex);

    _____
    ergebnis.toFirst();
    entfernungen.toFirst();
    while(ergebnis.hasAccess()) {
        Vertex v = _____
        double entfernungBisV = _____
        List<Vertex> nachbarn = karte.getNeighbours(v);
        for(nachbarn.toFirst();nachbarn.hasAccess();nachbarn.next()) {
            Vertex n = _____
            Edge e = _____
            double gewicht = _____
            if(_____ ) {
                ergebnis.append(n);
                _____
            }
        }
        ergebnis.next();
        entfernungen.next();
    }
    return ergebnis;
}

```

Verständnisfragen

1. Im „normalen“ Breitendurchlauf wird durch Markieren der Knoten verhindert, dass die `ergebnis`-Liste endlos weiterwächst.
Wie wird das in der vorliegenden Implementierung verhindert?
2. Auch in dieser Methode könnte man ein mehrfaches Besuchen der Knoten verhindern, indem man sie markiert, sobald sie in die `ergebnis`-Liste eingefügt werden und dann nicht mehr besucht.
Zeige anhand eines geeigneten Beispiel-Graphen auf, warum das für den Zweck der Methode (d.h. alle Knoten finden, die in `pMaxKm` zu erreichen sind) schädlich ist.