



### Bestimmung des optimalen Pfades - Breitensuche mit Knotenbewertung

- relativ speicherplatzsparend im Vgl. zur BS mit Speicherung aller aktiven Pfade
- wenig Zyklen nötig (im Vgl. zur Tiefensuche)
- Erweiterung durch heuristische Suche nach A\*-mgl

A = Menge der aktiven Knoten  
B(k) = Bewertung eines Knotens, d.h. die Länge des optimalen Pfades vom Startknoten bis k ;  
(die Länge eines Pfades ist seine Kantenanzahl)  
K = gesamte Knotenmenge  
G = int-Konstante mit  $G > \text{card}(K)$

---

**Init:** für alle  $k \in K$ :  $B(k) := G$   
A := {Startknoten}  
B(Startknoten) := 0

**Loop:** while  $A \neq \emptyset$  and Zielknoten  $\notin A$  do  
    wähle ein k aus A mit  $k = \arg \min_{x \in A} B(x)$  // minimaler aktiver Kn  
    A := A / {k} // Deaktivieren von k  
    N = {x ∈ K | x ist erreichbar von k} // mgl. Nachfolger von k  
    für alle n ∈ N: if B(n) == G then // n schon bewertet?  
        B(n) := B(k) + 1 // Bewerten von n  
        A := A ∪ {n} // Aktivieren von n  
    fi  
od

**Out:** if Zielknoten ∈ A then  
    optimalen Pfad aus B() vom Zielknoten beginnend rückwärts auslesen  
    ( Abstieg entlang der Kanten in Richtung fallender Bewertung)  
else  
    Zielknoten vom Startknoten nicht erreichbar  
fi

---

nötiger Speicher (Datenbasis):  
- Knotenarray der Aktivierungen  
- Knotenarray der Bewertungen