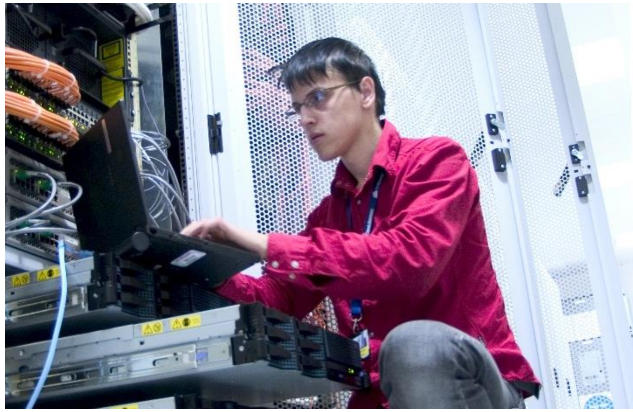
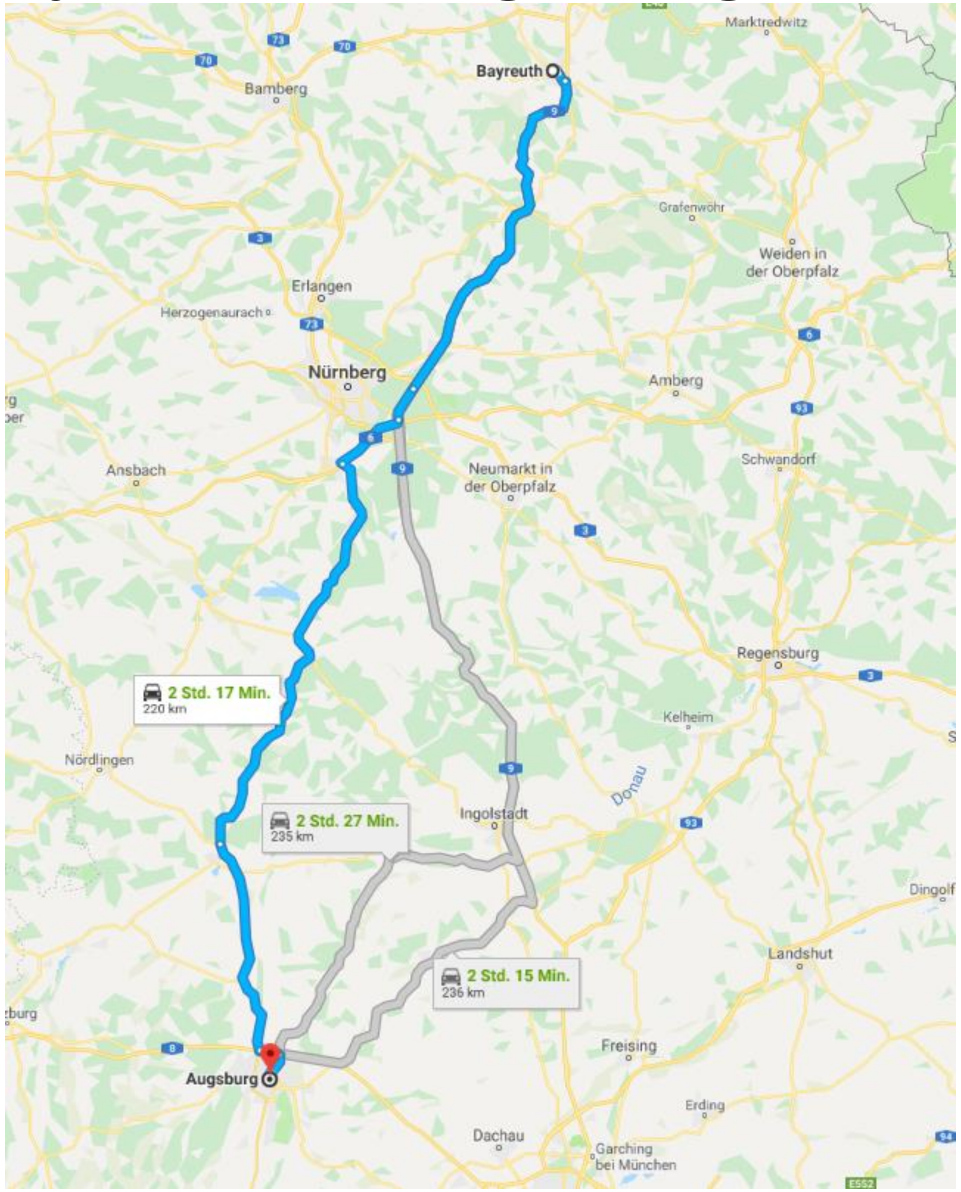
EMIL-VON-BEHRING GYMNASIUM

Graphendurchlauf III – Dijkstra-Algorithmus

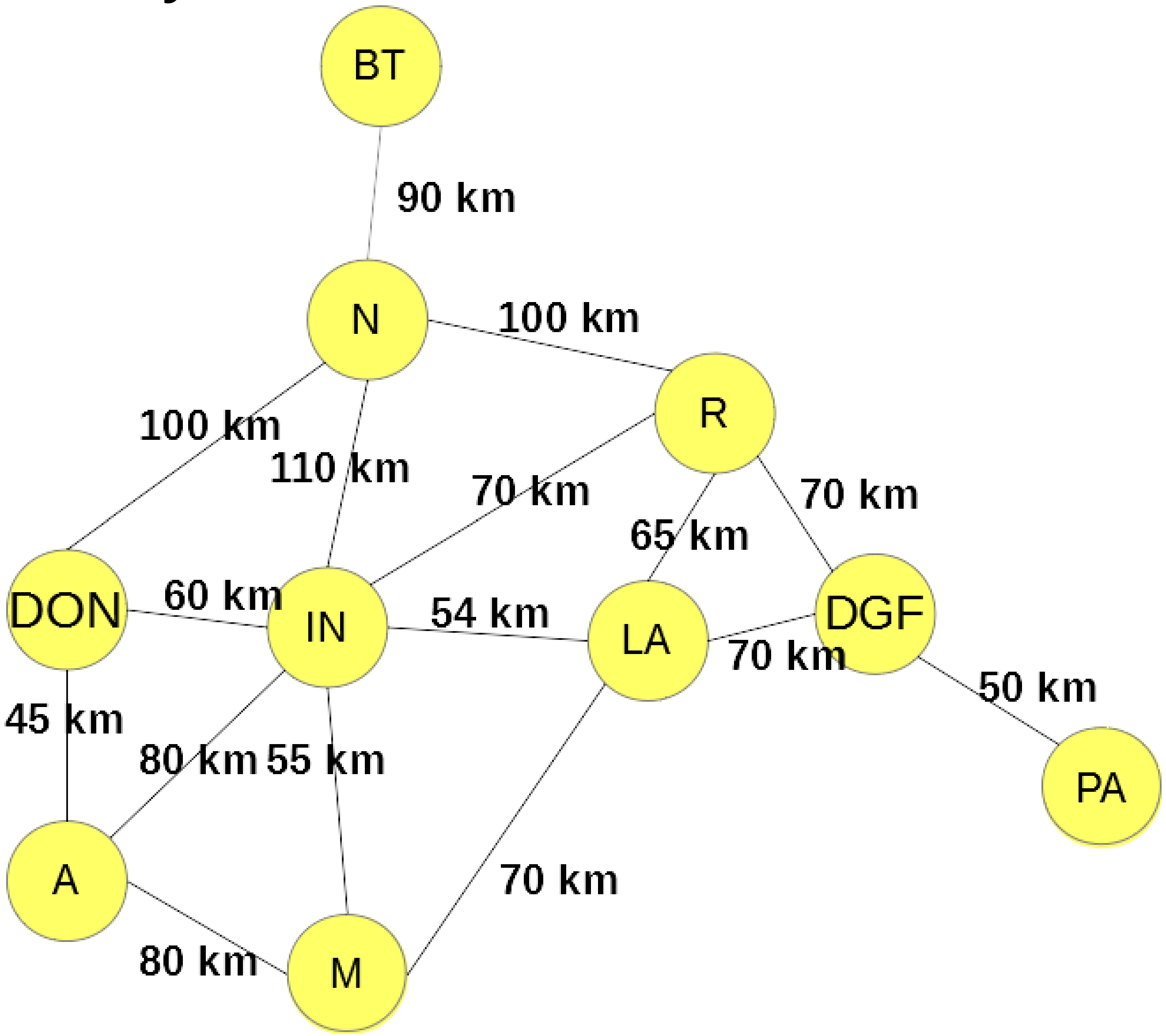
Kürzester Weg zwischen zwei Knoten



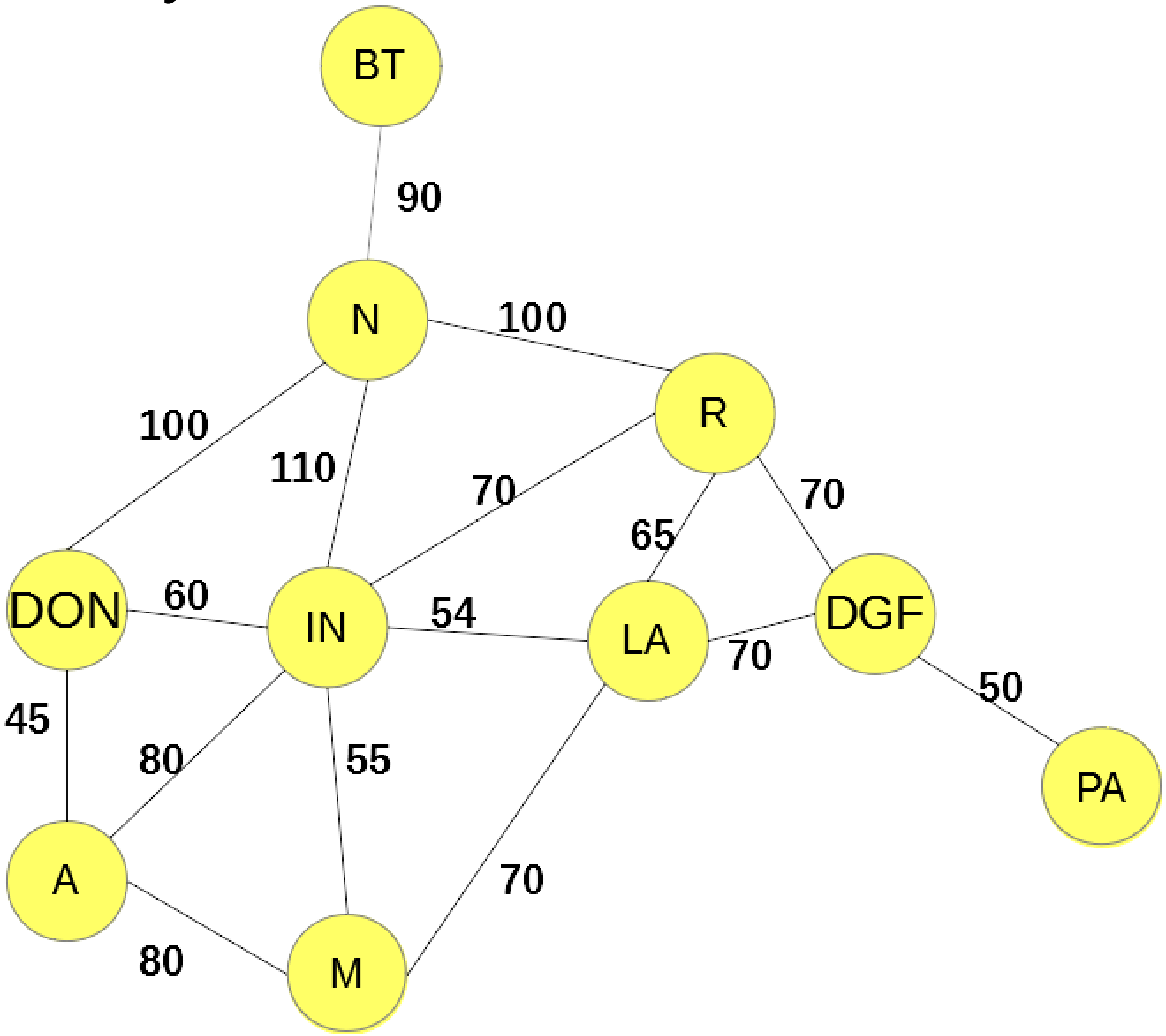
Kürzester Weg: Bayreuth - Augsburg



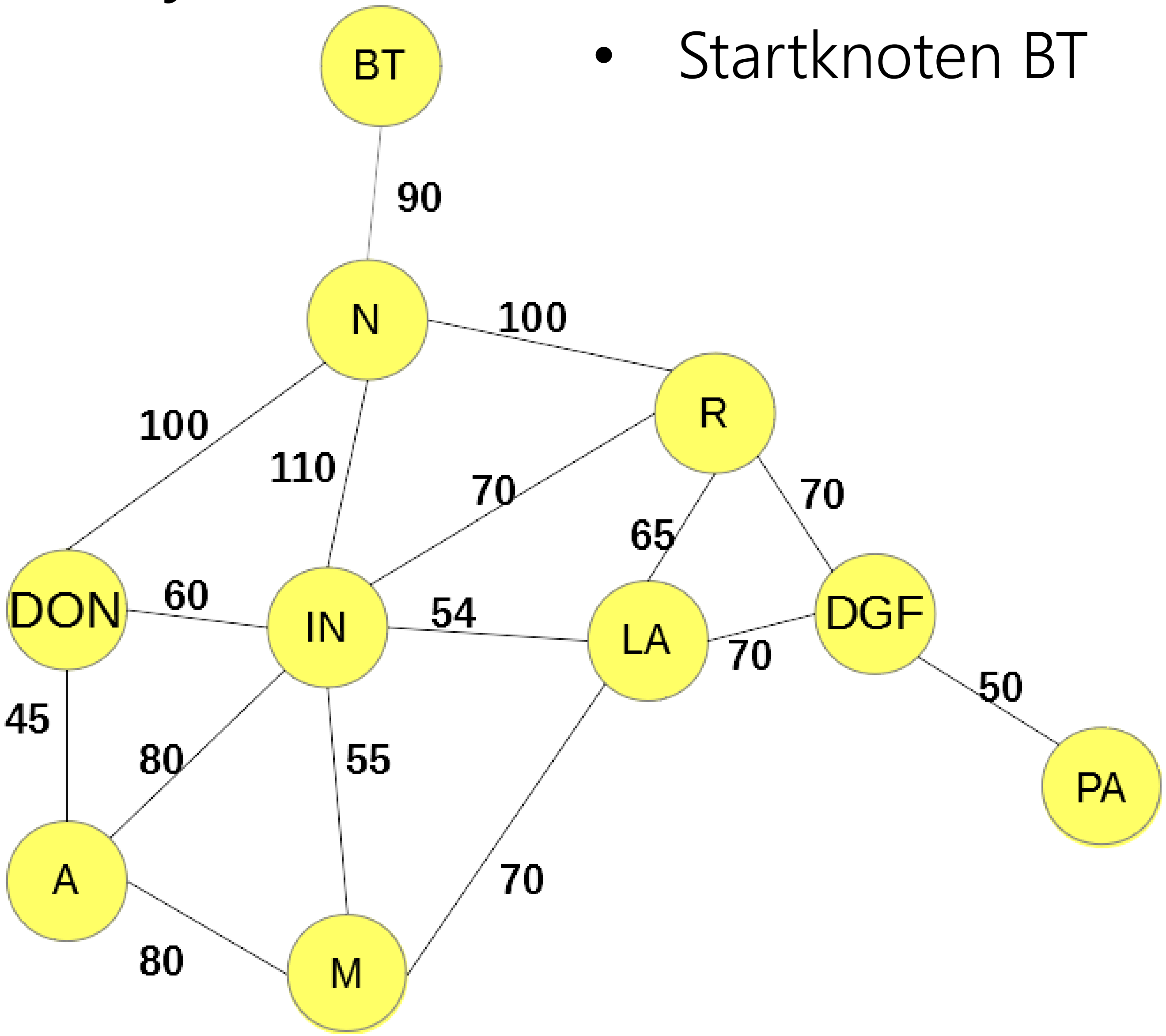
Autobahnnetz Bayern



Autobahnnetz Bayern

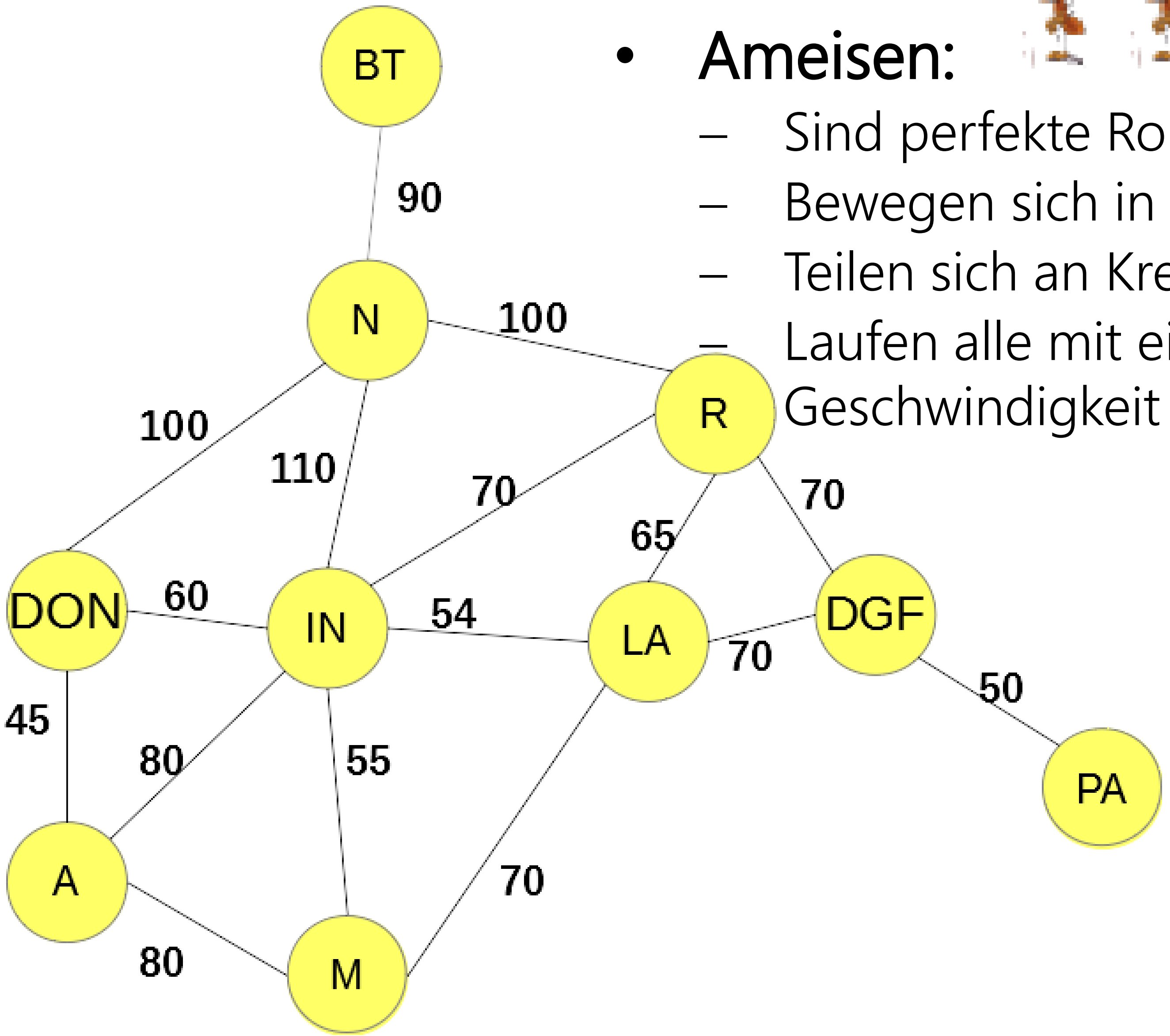


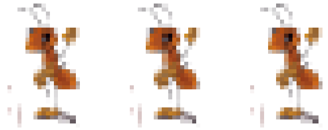
Autobahnnetz Bayern



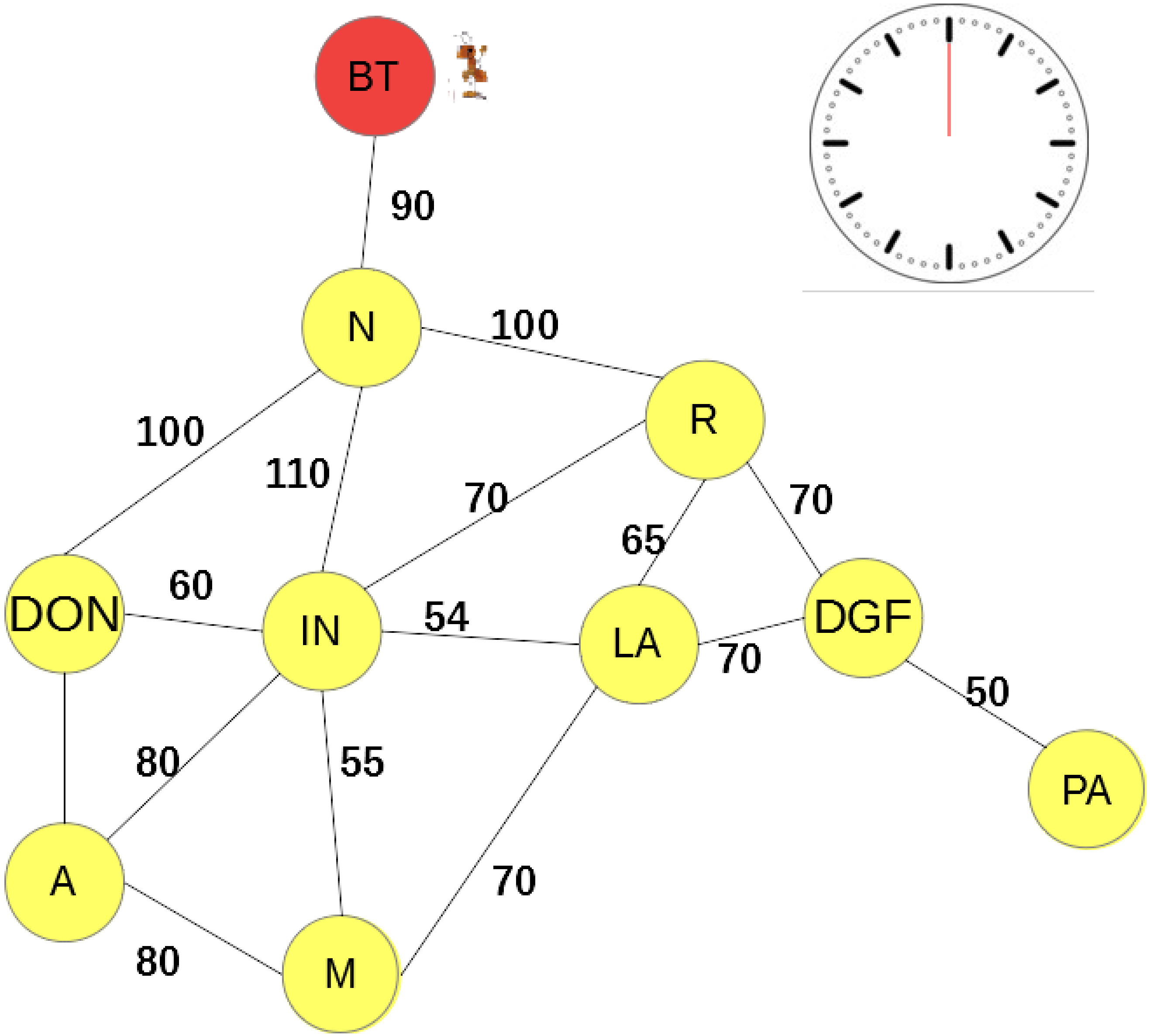
- Startknoten BT

Ein Blick in die Natur

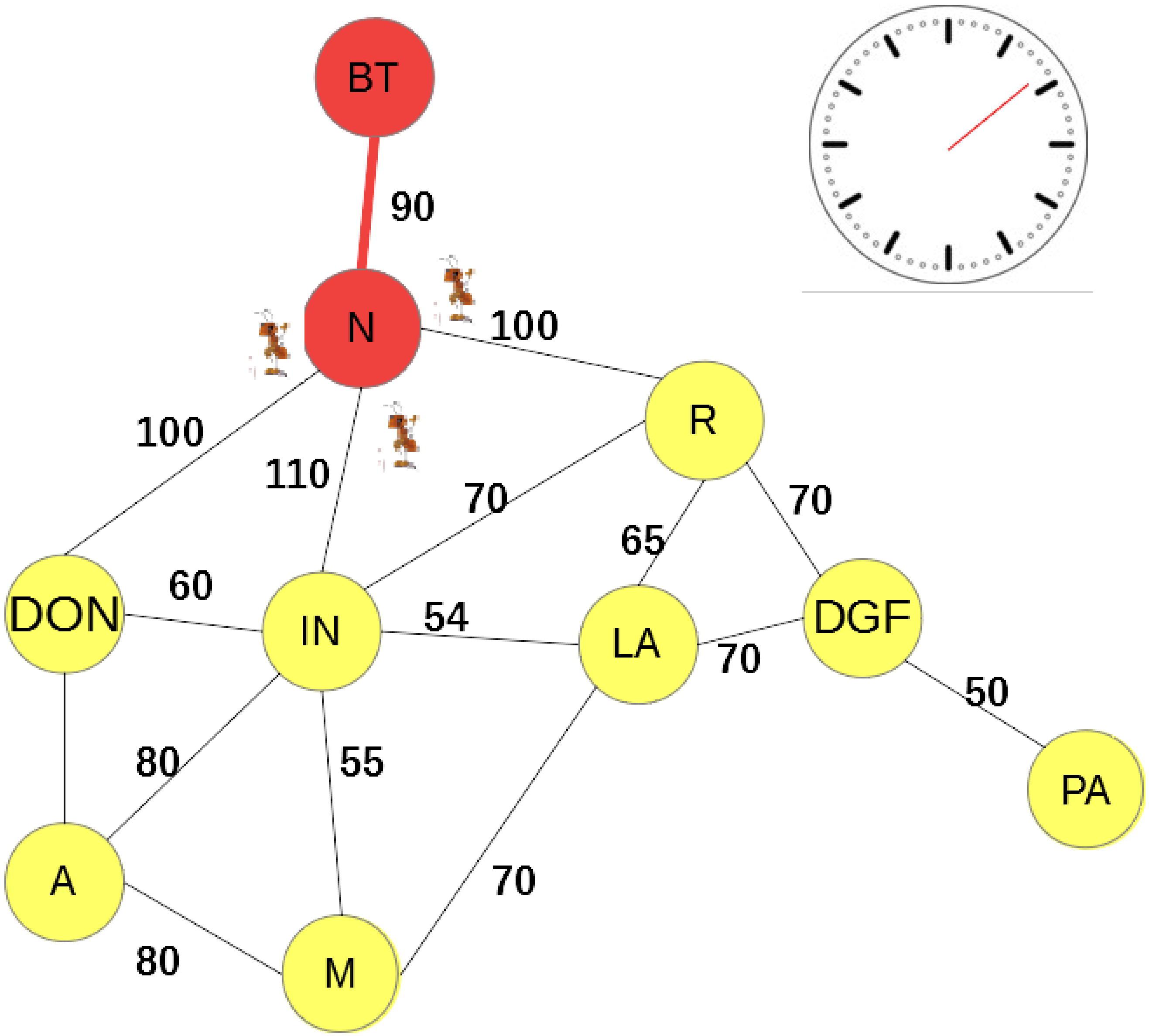


- **Ameisen:** 
  - Sind perfekte Routenplaner
  - Bewegen sich in Gruppen
  - Teilen sich an Kreuzungspunkten auf
  - Laufen alle mit einer konstanten Geschwindigkeit

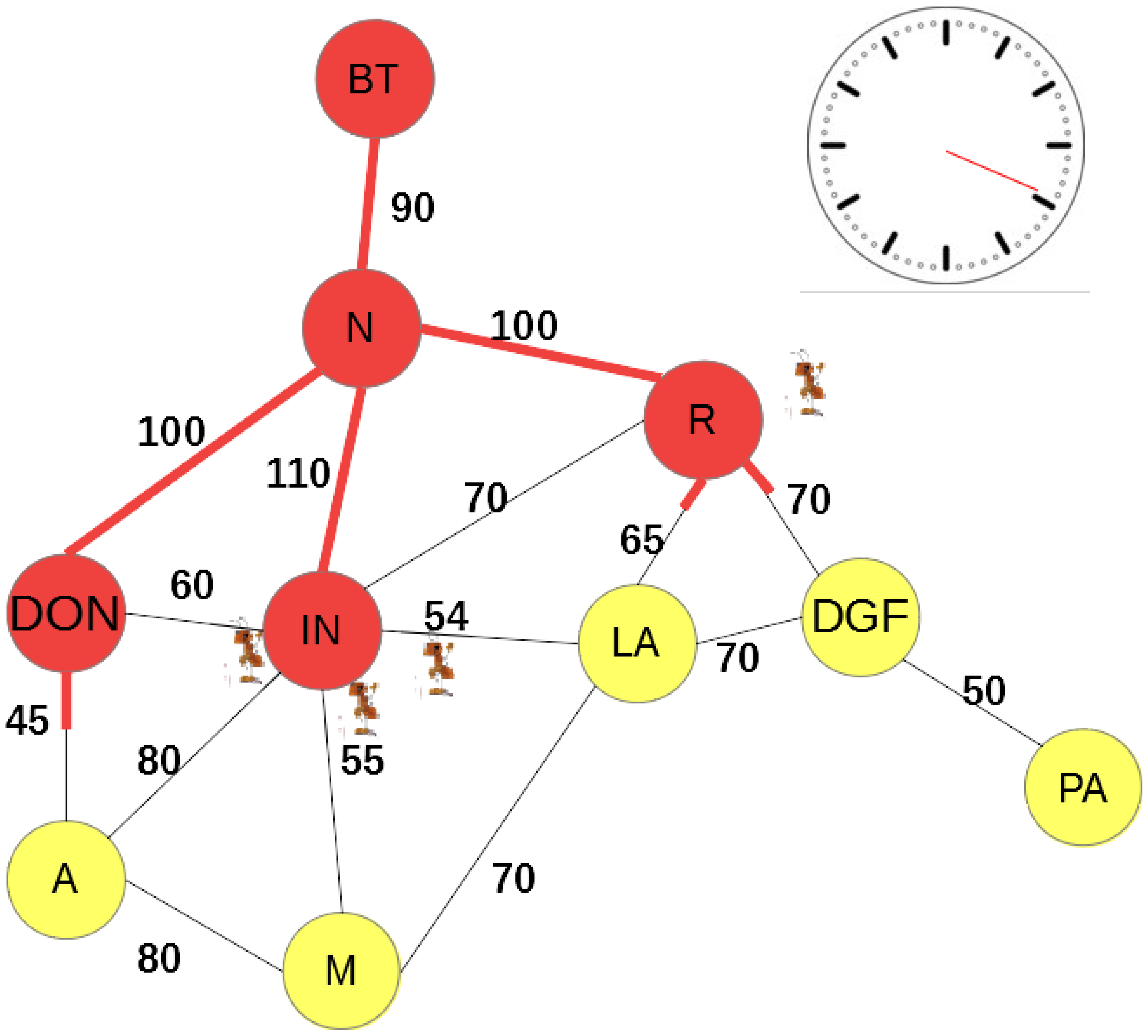
Ein Blick in die Natur



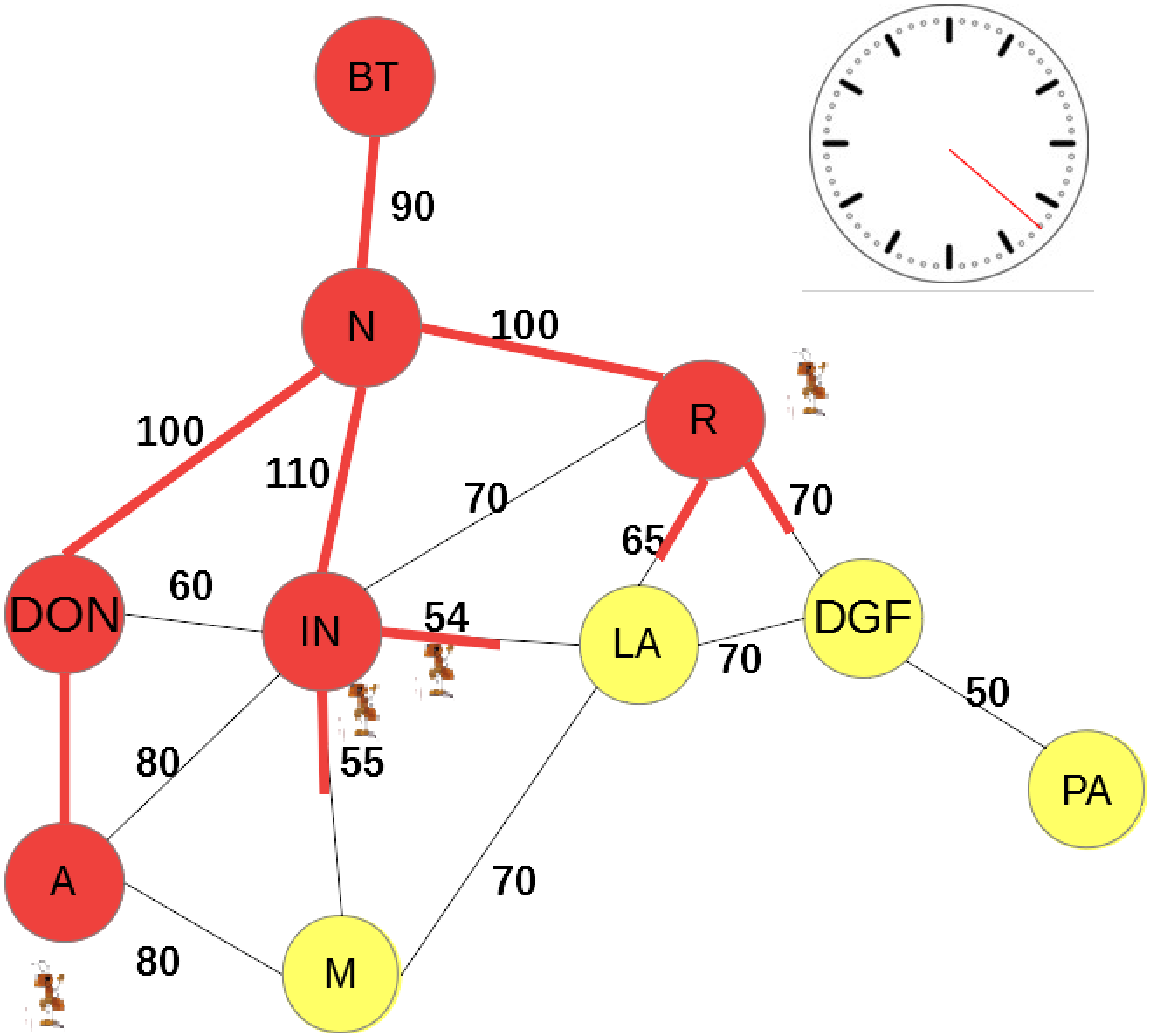
Ein Blick in die Natur



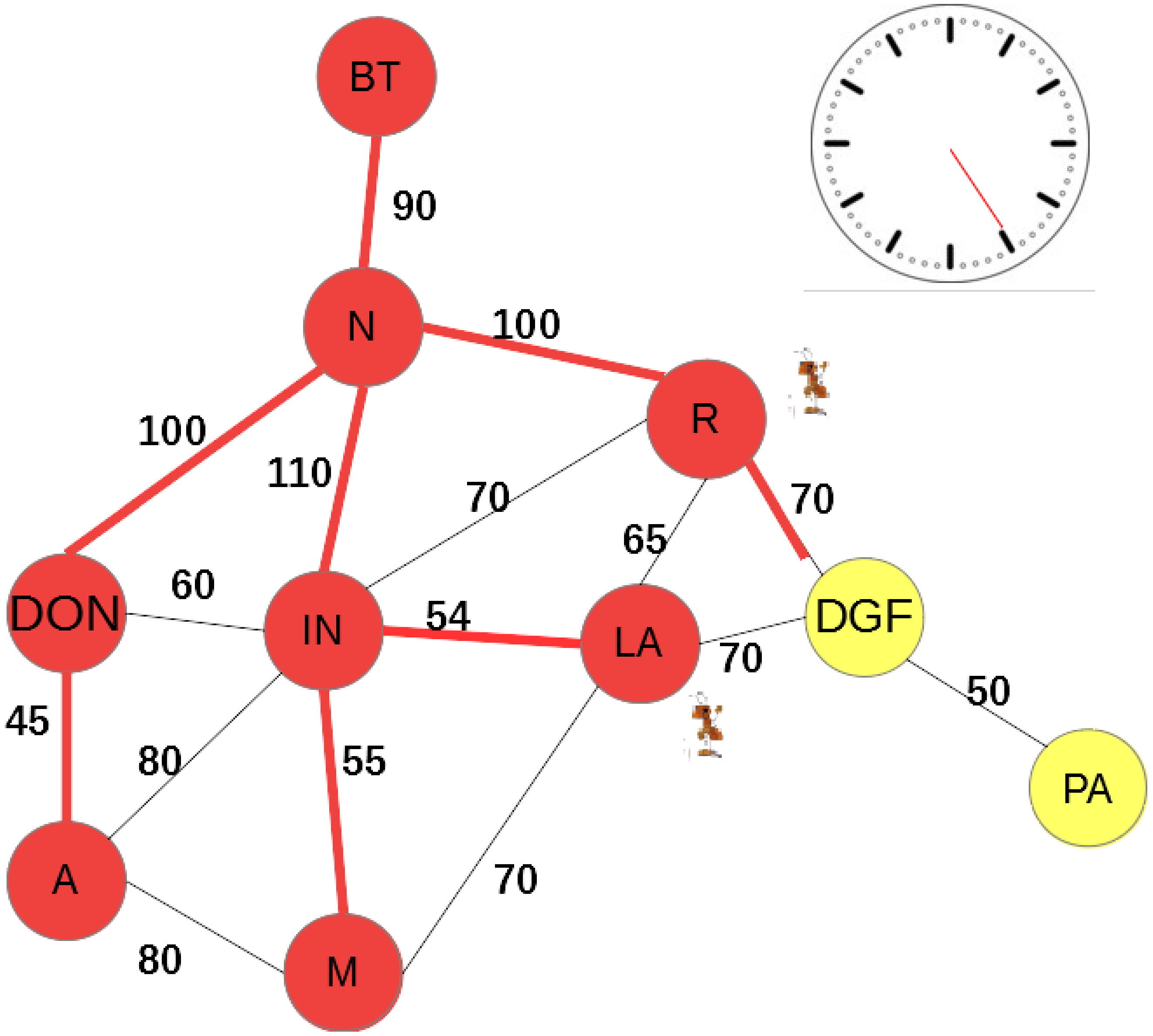
Ein Blick in die Natur



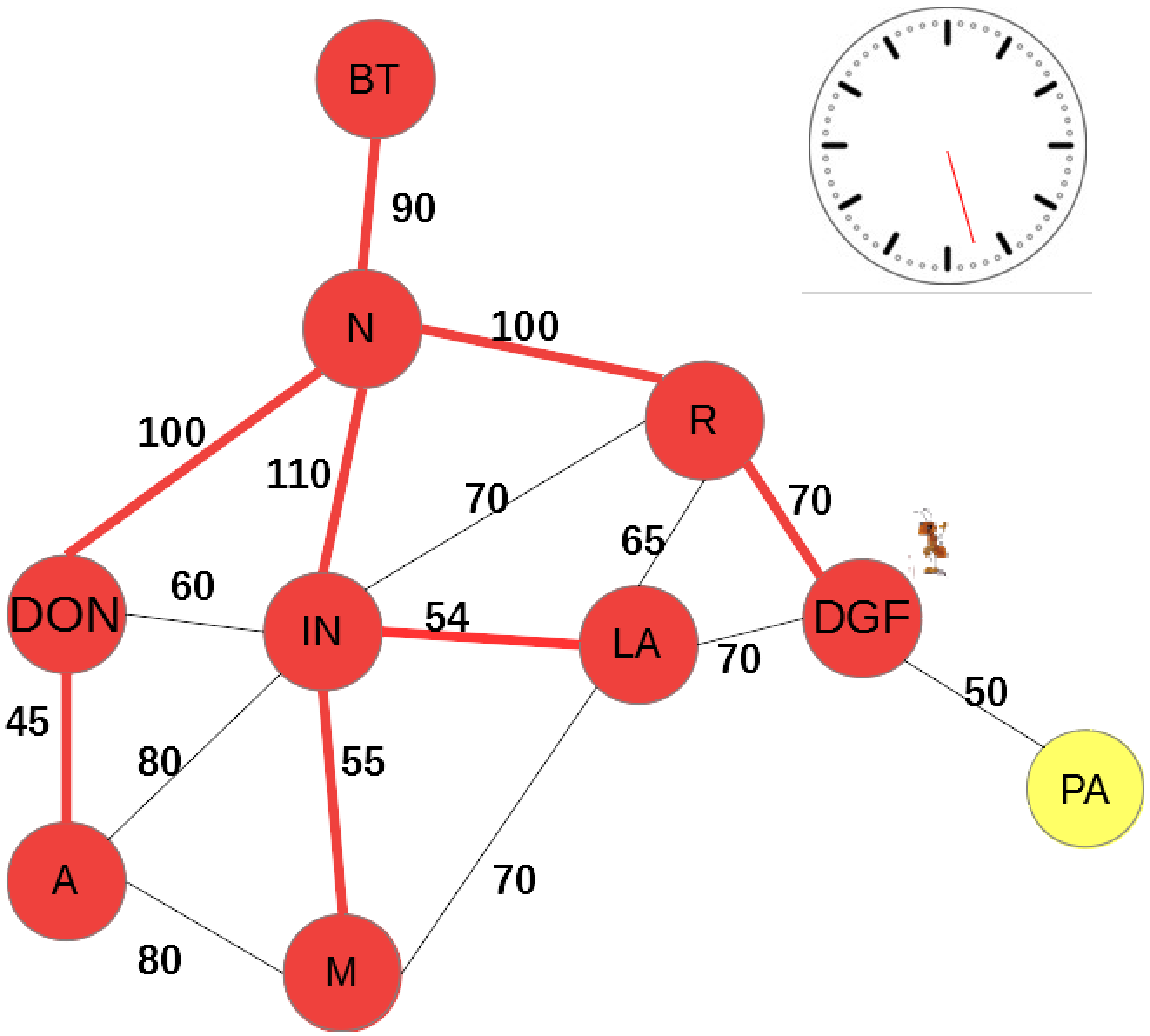
Ein Blick in die Natur



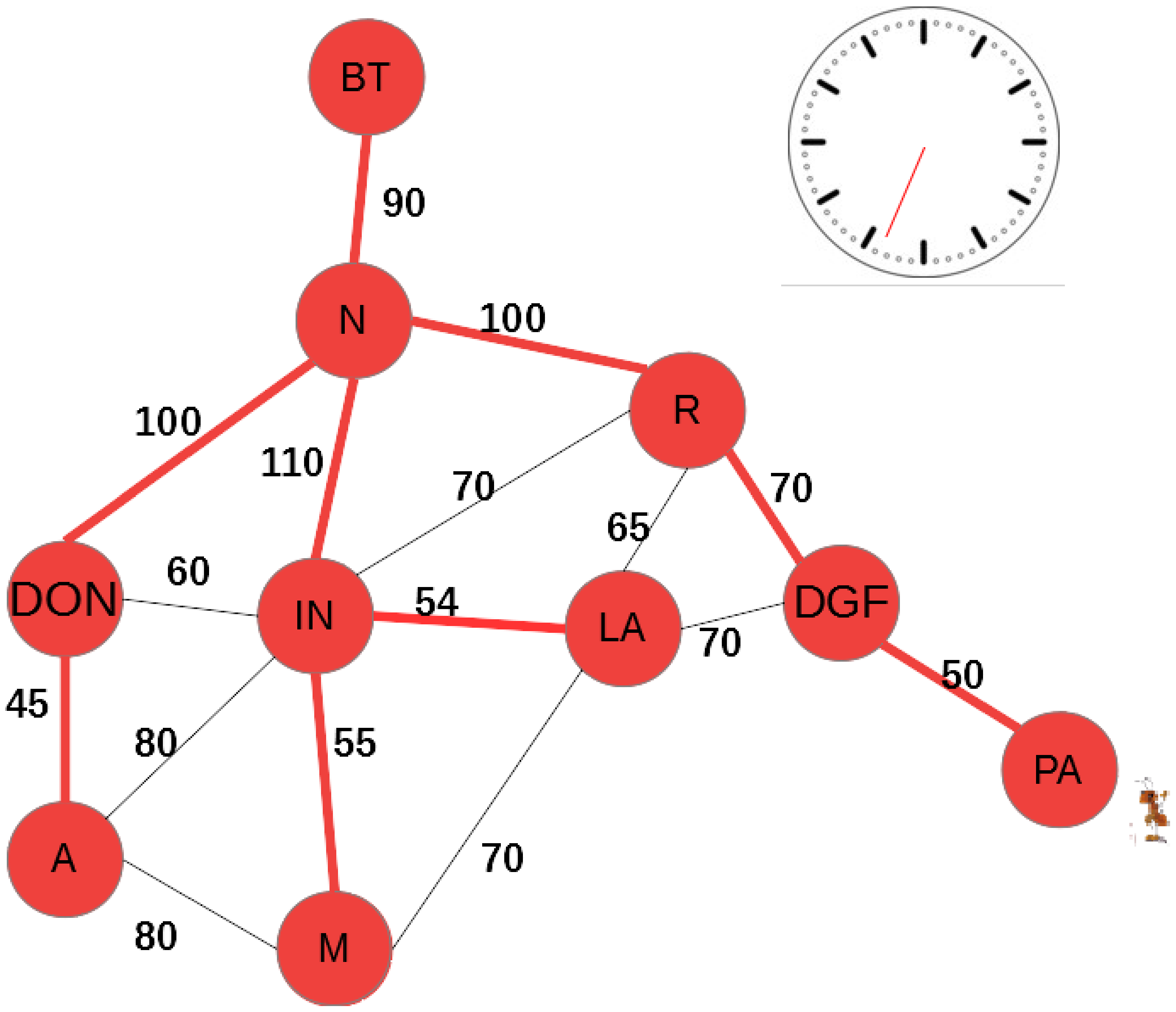
Ein Blick in die Natur



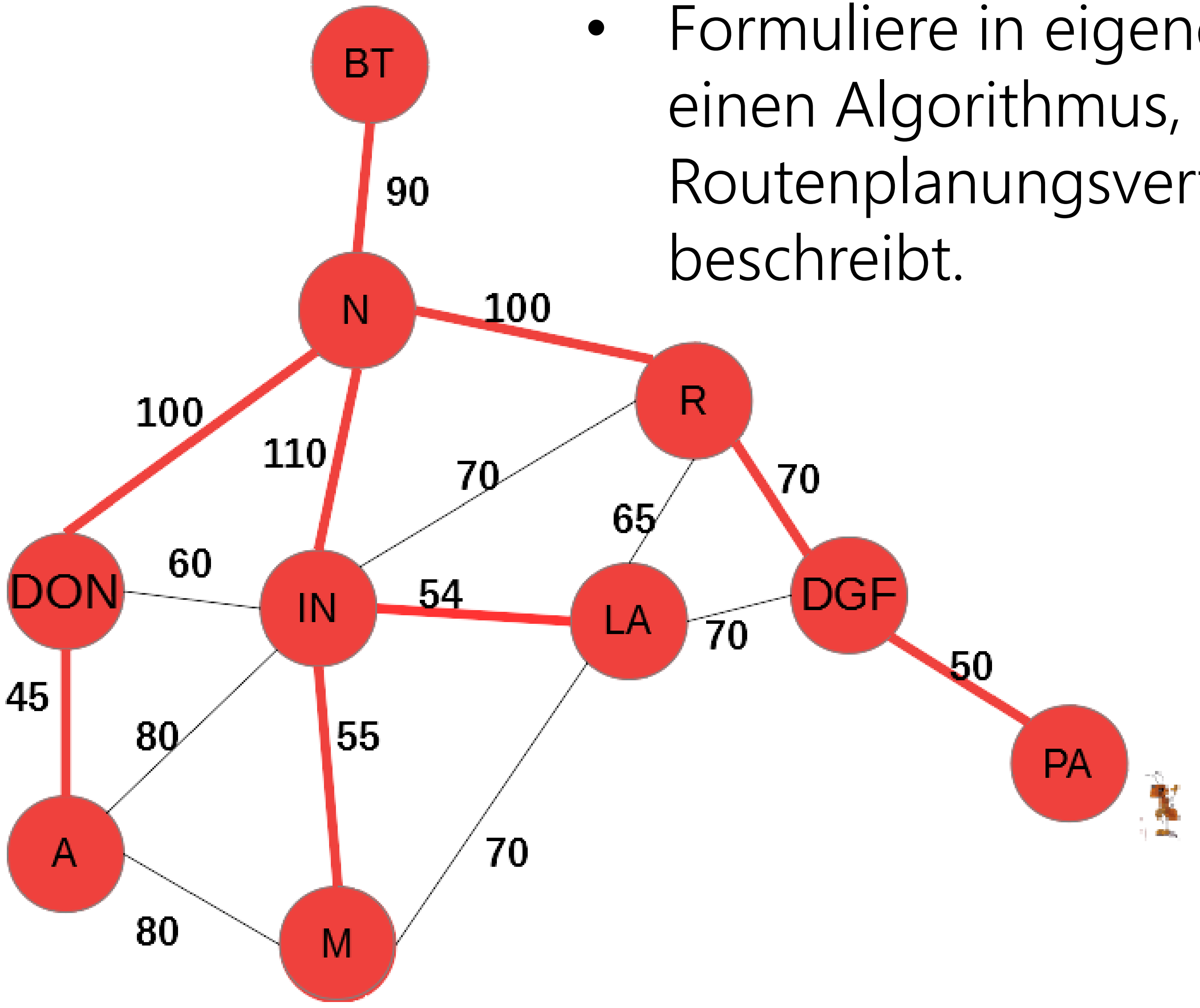
Ein Blick in die Natur



Ein Blick in die Natur



Ein Blick in die Natur



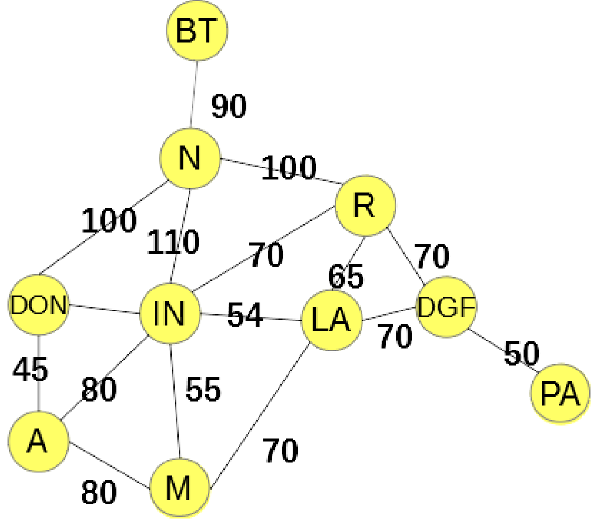
- Formuliere in eigenen Worten einen Algorithmus, der das Routenplanungsverfahren beschreibt.

Der Dijkstra-Algorithmus

- Markiere den Startknoten, weise ihm den Gesamtaufwand 0 zu, verwende ihn als aktuellen Knoten.
- Untersuche alle Nachbarknoten des aktuellen Knotens:
  – Berechne den Gesamtaufwand des Nachbarknotens ausgehend vom aktuellen Knoten
  – Ist der Nachbarknoten vom aktuellen Knoten mit einer kürzeren Distanz erreichbar, so wird diese Distanz aktualisiert/neu gesetzt, falls noch keine Distanz gesetzt ist.
- Wähle einen noch nicht besuchten Knoten mit dem geringsten Gesamtaufwand. Dieser ist der neue aktuelle Knoten.
- Fahre so lange mit Punkt 2 fort, bis der Zielknoten erreicht ist.

Der Dijkstra-Algorithmus

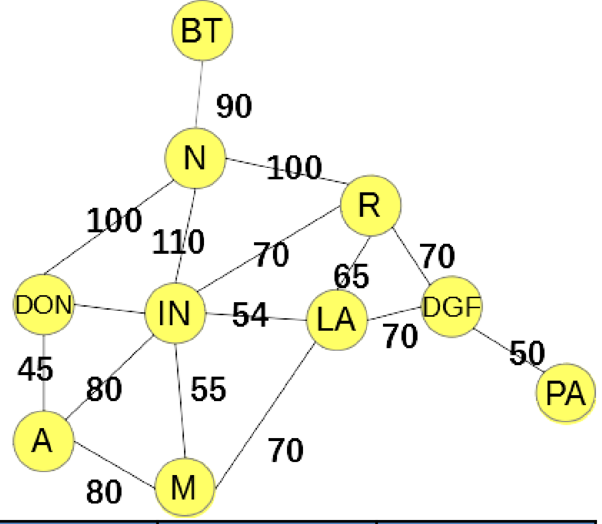
- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.



Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞

Der Dijkstra-Algorithmus

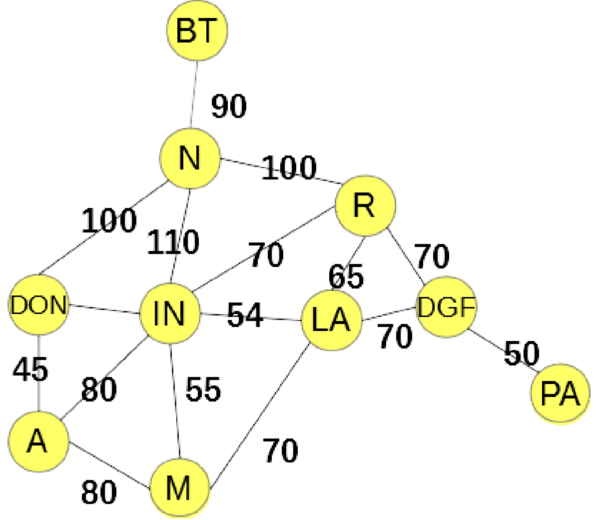
- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.



Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞

Neuer aktueller Knoten

Der Dijkstra-Algorithmus

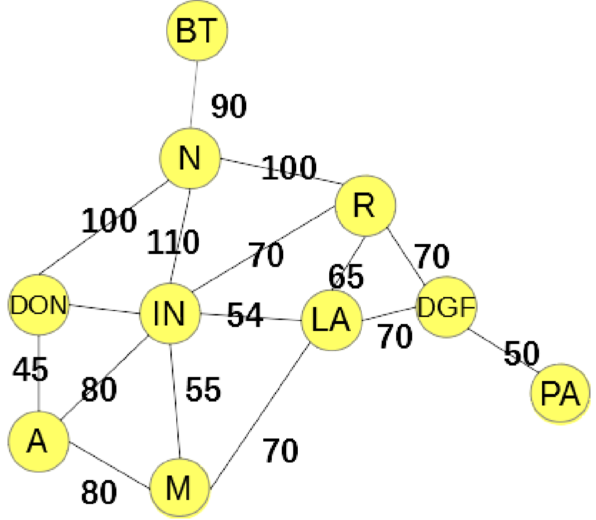


- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞

R, IN und DON können nun erreicht werden.

Der Dijkstra-Algorithmus

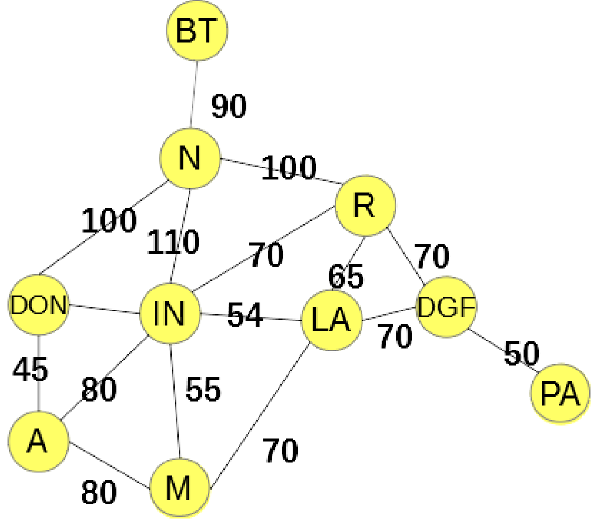


- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞

Schreibe immer den Gesamtaufwand ausgehend vom Startknoten auf!

Der Dijkstra-Algorithmus

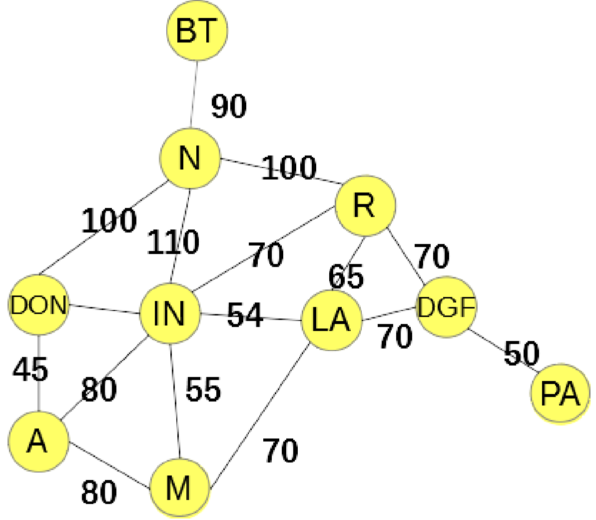


- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞

Check: IN kann auf von R mit 90+100+70=260 erreicht werden. Da 260>200 gilt, bleibt die kürzeste Entfernung bei 200 über N. Wäre sie kürzer müsste dieses Feld neu gesetzt werden

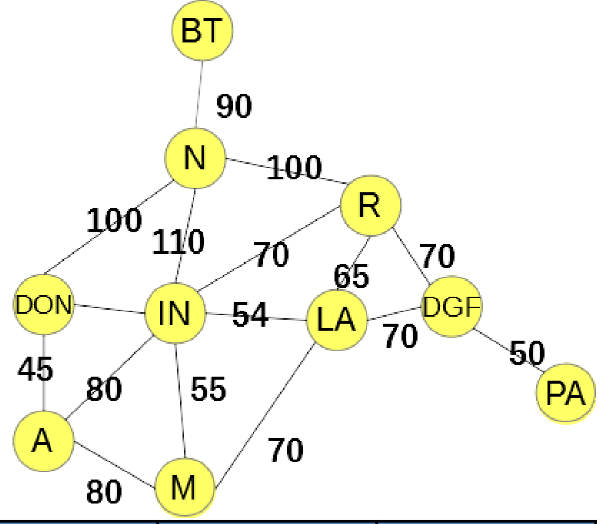
Der Dijkstra-Algorithmus



- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞

Der Dijkstra-Algorithmus

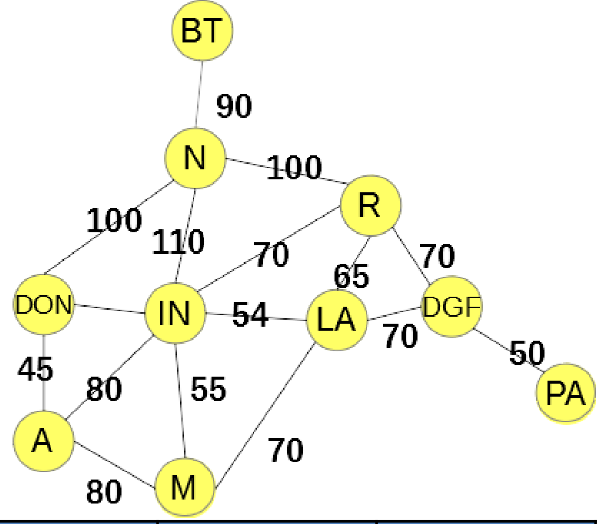


- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞
DON,IN				-		254,IN	260,R	235,DON	255,IN	∞

Achtung Update: LA ist über IN schneller zu erreichen als auf den bisherigen Weg, weil 254<255!

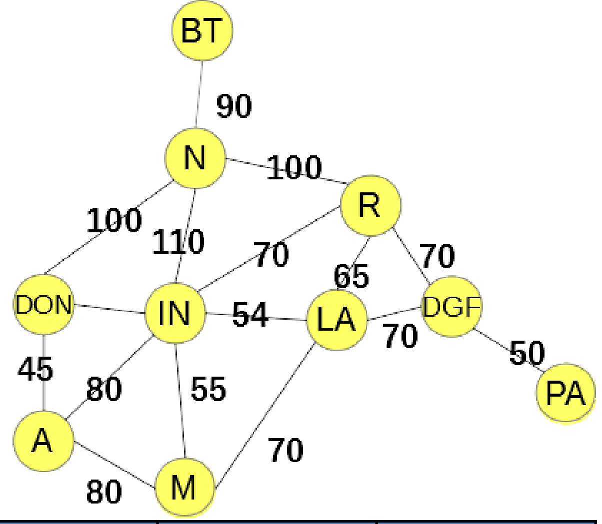
Der Dijkstra-Algorithmus



- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞
DON,IN				-		254,IN	260,R	235,DON	255,IN	∞
…,IN,A						254,IN	260,R	-	255,IN	∞

Der Dijkstra-Algorithmus

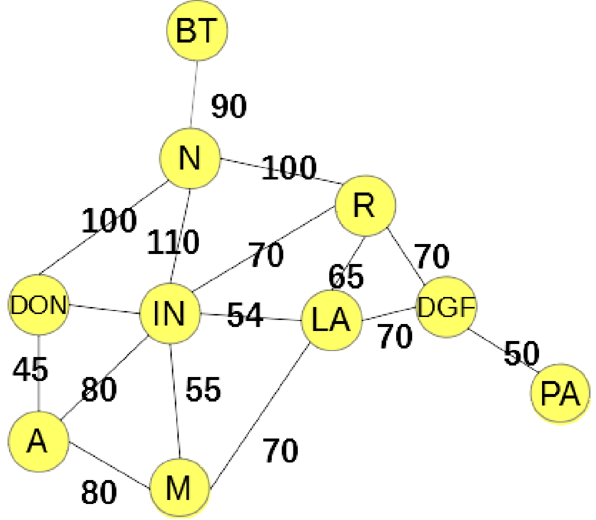


- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞
DON,IN				-		254,IN	260,R	235,DON	255,IN	∞
…,IN,A						254,IN	260,R	-	255,IN	∞
…,A,LA						-	260,R		255,IN	∞

Der Dijkstra-Algorithmus

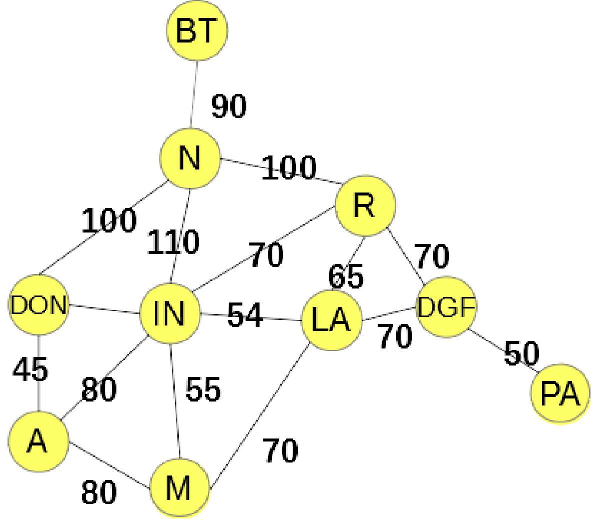
- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.



Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞
DON,IN				-		254,IN	260,R	235,DON	255,IN	∞
…,IN,A						254,IN	260,R	-	255,IN	∞
…,A,LA						-	260,R		255,IN	∞
…,LA,M							260,R		-	∞

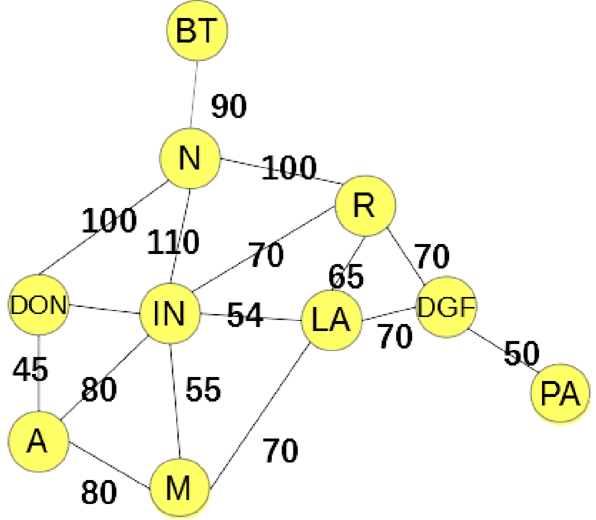
Der Dijkstra-Algorithmus

- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.



Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞
DON,IN				-		254,IN	260,R	235,DON	255,IN	∞
...,IN,A						254,IN	260,R	-	255,IN	∞
...,A,LA						-	260,R		255,IN	∞
...,LA,M							260,R		-	∞
,M,DGF							-			310,DGF

Der Dijkstra-Algorithmus



- d(K) = Abstand zum Knoten
- V = Vorheriger Knoten der die aktuell kürzeste Distanz zum Knoten besitzt
- Bsp: d(D), V = 4,F bedeutet: Knoten D wird von Knoten F mit Gesamtaufwand 4 erreicht.

Knoten	d(BT),V	d(N),V	d(R),V	d(IN),V	d(DON),V	d(LA),V	d(DGF),V	d(A),V	d(M),V	d(PA),V
BT	-	90,BT	∞	∞	∞	∞	∞	∞	∞	∞
BT,N		-	190,N	200,N	190,N	∞	∞	∞	∞	∞
BT,N,R			-	200,N	190,N	255,R	260,R	∞	∞	∞
..,R,DON				200,N	-	255,R	260,R	235,DON	∞	∞
DON,IN				-		254,IN	260,R	235,DON	255,IN	∞
...,IN,A						254,IN	260,R	-	255,IN	∞
...,A,LA						-	260,R		255,IN	∞
...,LA,M							260,R		-	∞
,M,DGF							-			310,DGF
DGF,PA										-

Übung Dijkstra

- Siehe Skript Aufgabe Dijkstra-Algorithmus