

# Künstliche Intelligenz



# Künstliche Intelligenz (2)

- Definition
  - Der Versuch, spezifische geistige Fähigkeiten des Menschen maschinell nachzuahmen.

# Kennzeichen von Intelligenz

- Wahrnehmung
- Logisches Denken
- Lernen
- Kommunikation
- Agieren

# Beispiel Turing-Test

- Frage: kann eine Maschine menschliche Kommunikation simulieren?
- Bis heute ist man weit davon entfernt, dieses Ziel zu erreichen.
- ELIZA

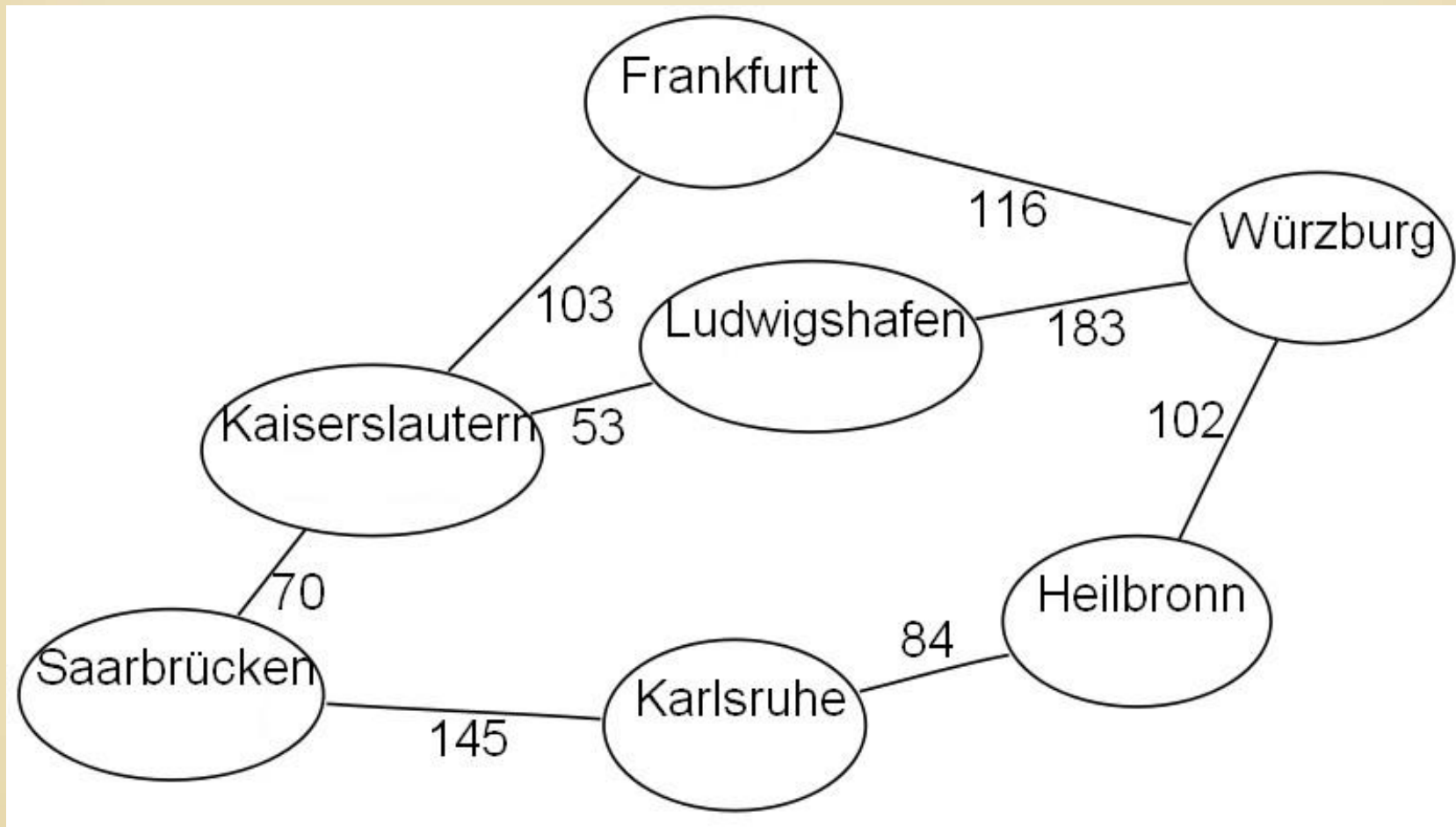
# Anwendungen

- Mustererkennung
  - automatische Erkennung von Handschrift, Stimme, Gesichtern usw.
- Robotersteuerung
- Computerspiele
  - Schach, Strategiespiele, Simulationen
- Expertensysteme
  - z.B. medizinische Diagnose
- Und viele mehr...

# Graphen und Bäume

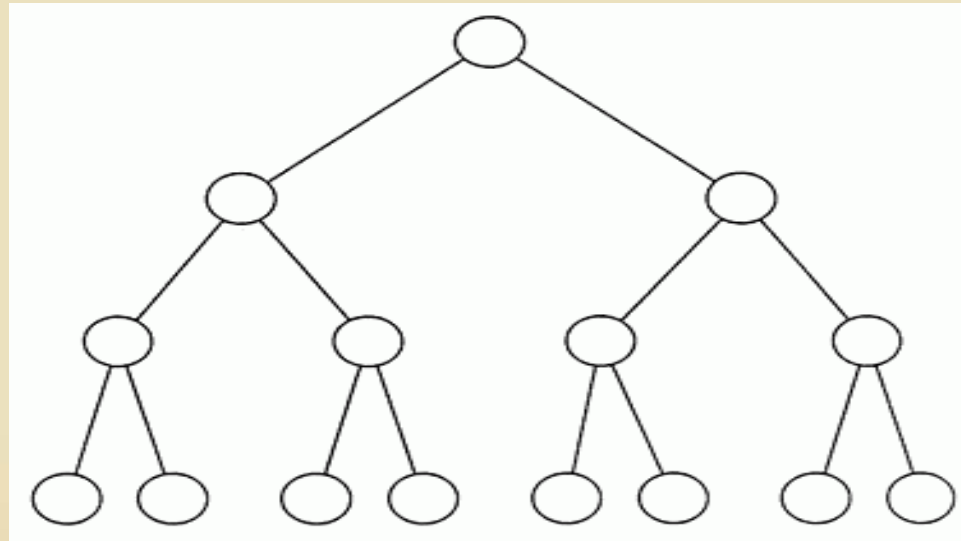
- Viele Probleme der realen Welt können durch Graphen dargestellt werden
- Ein Graph besteht aus:
  - Knoten
  - Kanten

# Beispiel: Städte mit Entfernungen



# Bäume

- Speziellfall eines Graphen
- Haben eine Wurzel
- Jeder Knoten hat genau einen Vorgänger
- Bsp.:

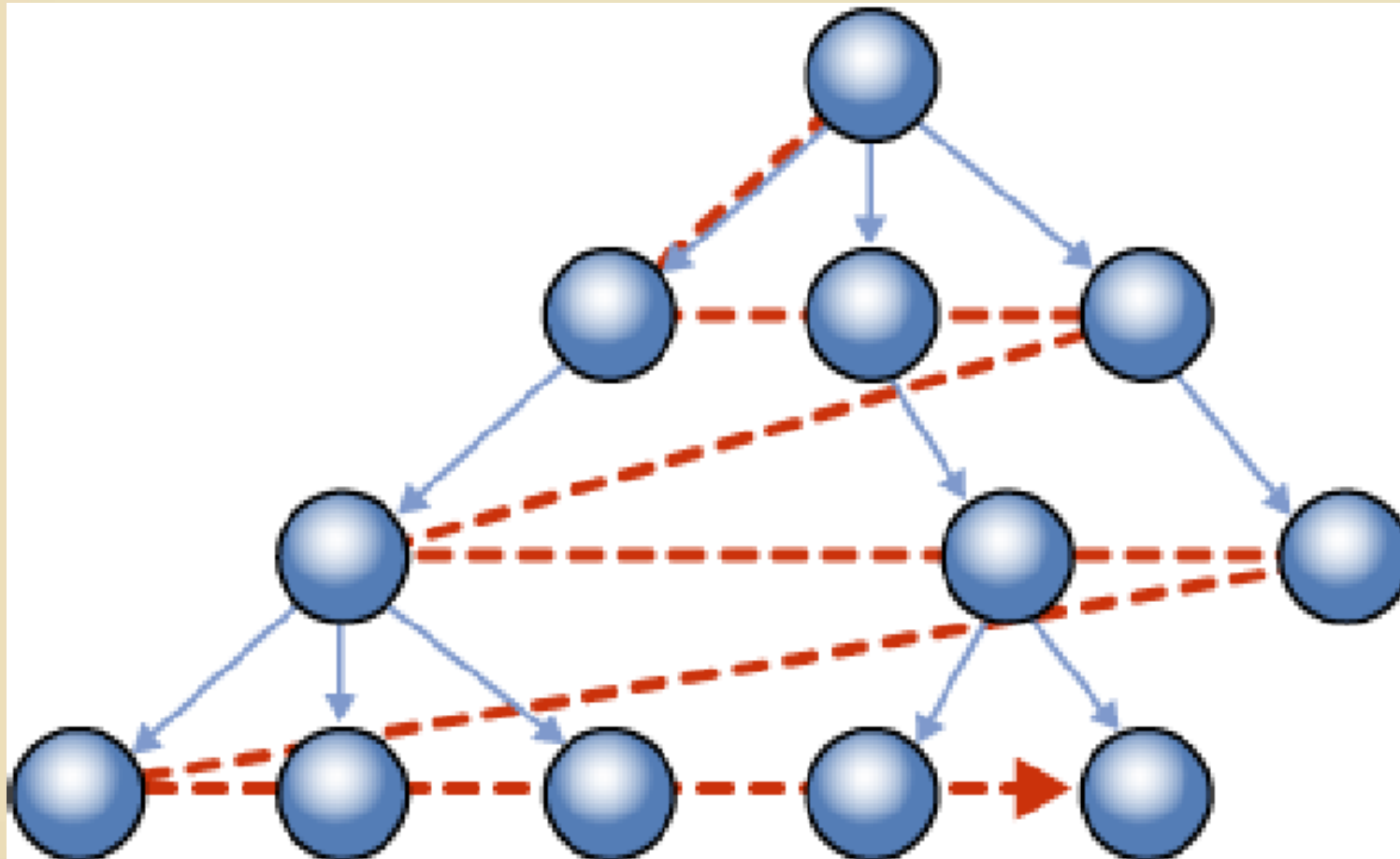




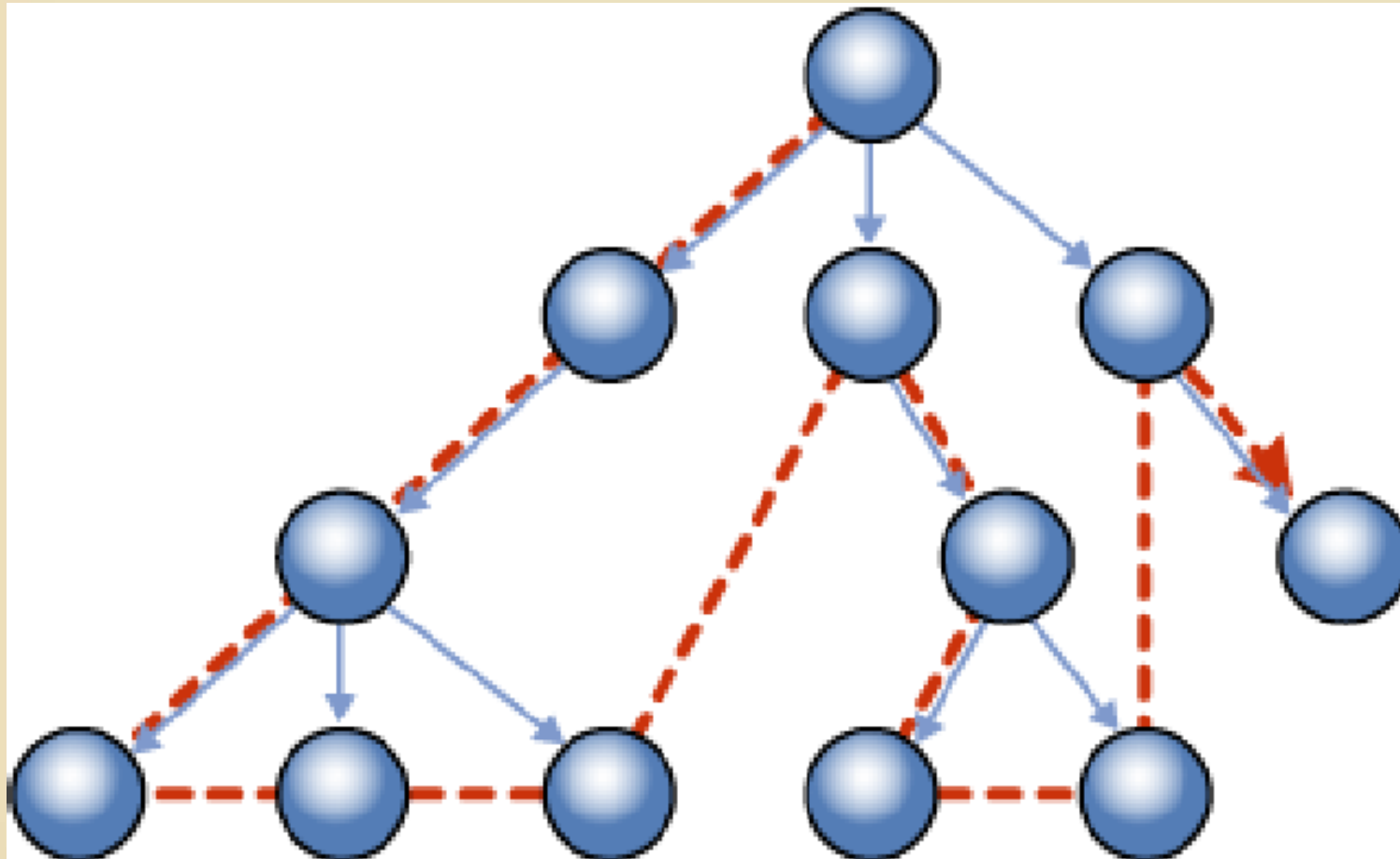
# Baumsuche

- Viele Probleme können mit Bäumen modelliert werden
- -> Baumsuche ist ein wichtiges Grundelement der KI
- Zwei wichtige Verfahren:
  - Tiefensuche
  - Breitensuche

# Breitensuche



# Tiefensuche



# A\*-Algorithmus

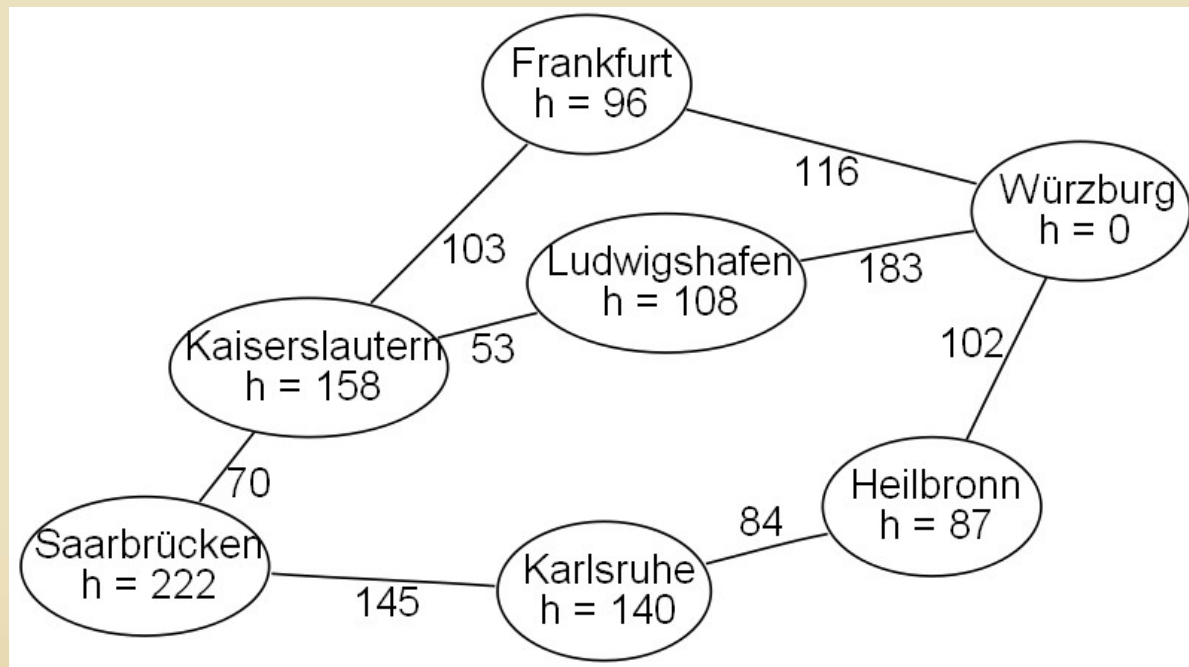
- Findet kürzesten Pfad zwischen zwei Knoten in einem Graphen
- Verwendung:
  - Routenplaner
  - Computerspiele
  - und weitere

# Grundidee

- Ähnlich wie Breiten- oder Tiefensuche
- Aber: Beschleunigung der Suche durch *Heuristik* (d.h. Abschätzung)
- Jedem Knoten  $N$  werden Kosten  $f(N)$  zugeordnet:  $f(N) = g(N) + h(N)$ 
  - $g(N)$ : Kosten vom Startknoten bis  $N$
  - $h(N)$ : Geschätzte Kosten von  $N$  bis zu zum Zielknoten

# Beispiel einer Heuristik

- Suche den kürzesten Weg von Saarbrücken nach Würzburg
- Heuristik  $h$ : Luftlinie zum Ziel



# Algorithmus in Worten

- Beginne mit Startknoten
- Berechne die Funktion  $f(n) = g(n) + h(n)$  für alle Nachfolgeknoten
- Wähle den Knoten mit dem kleinsten  $f(n)$  aus, bestimme seine Nachfolger und berechne ihre  $f(n)$
- Wiederhole diesen Vorgang bis der Zielknoten gefunden ist

# Algorithmus (Pseudocode) 1/2

- Zwei Listen sind dabei wichtig:
  - OPEN: entdeckte Knoten, deren Nachfolger noch ermittelt werden müssen
  - CLOSED: Knoten, deren Nachfolger bereits ermittelt wurden



# Algorithmus (Pseudocode) 2/2

**program** a-star

    OPEN.add(startNode)

    repeat

        bestNode := OPEN.remove(bestNode)

**if** (bestNode == targetNode) **then return** bestNode

        OPEN.add(bestNode.successors())

        CLOSED.add(bestNode)

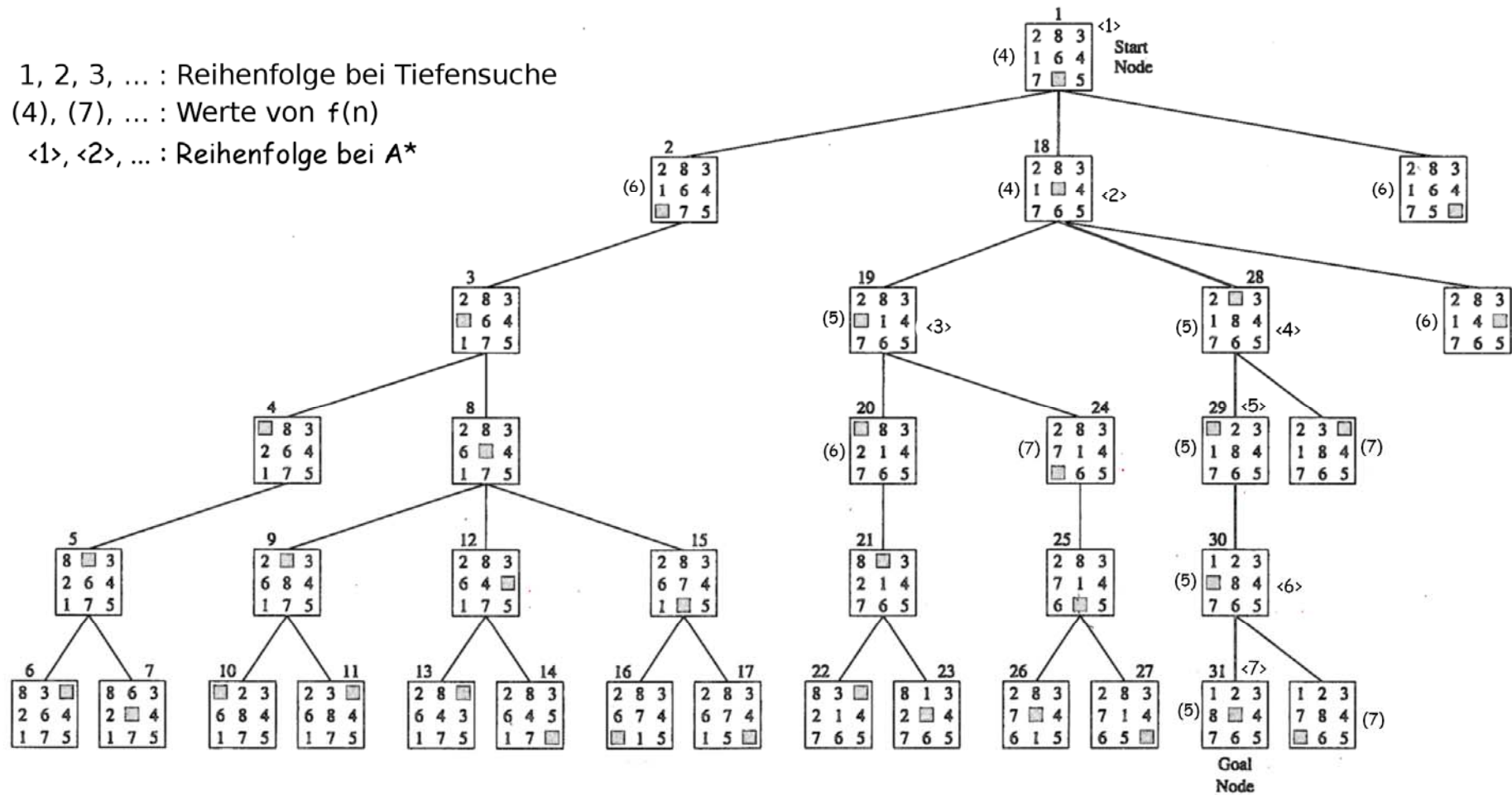
**until** (OPEN.isEmpty)

**return** no path found

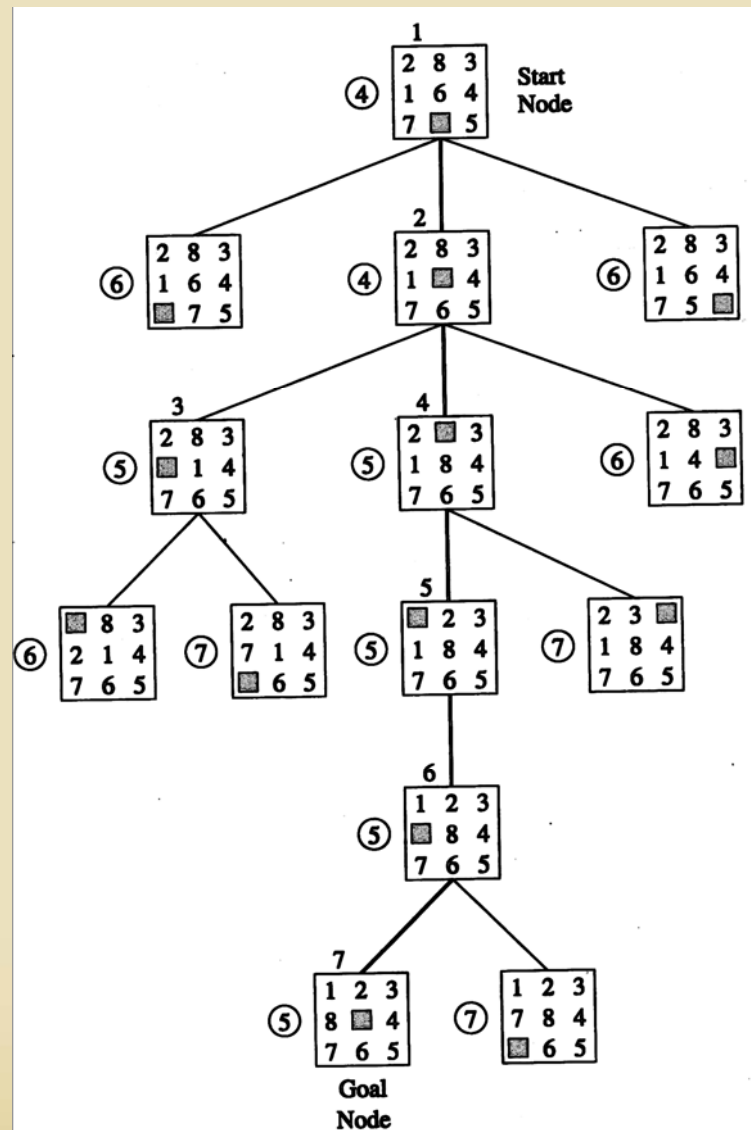
**end**

# Lösung zu Aufgabe 1)

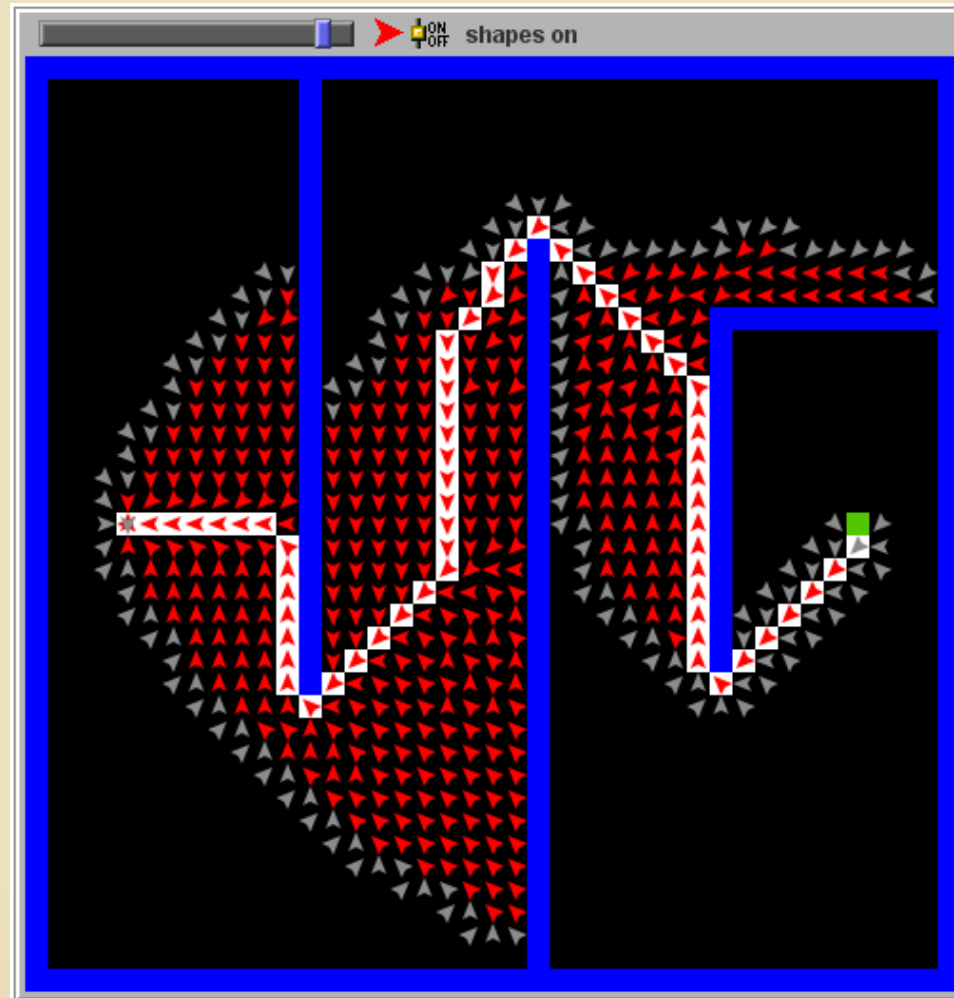
1, 2, 3, ... : Reihenfolge bei Tiefensuche  
 (4), (7), ... : Werte von  $f(n)$   
 <1>, <2>, ... : Reihenfolge bei  $A^*$



# Lösung zu Aufgabe 3)

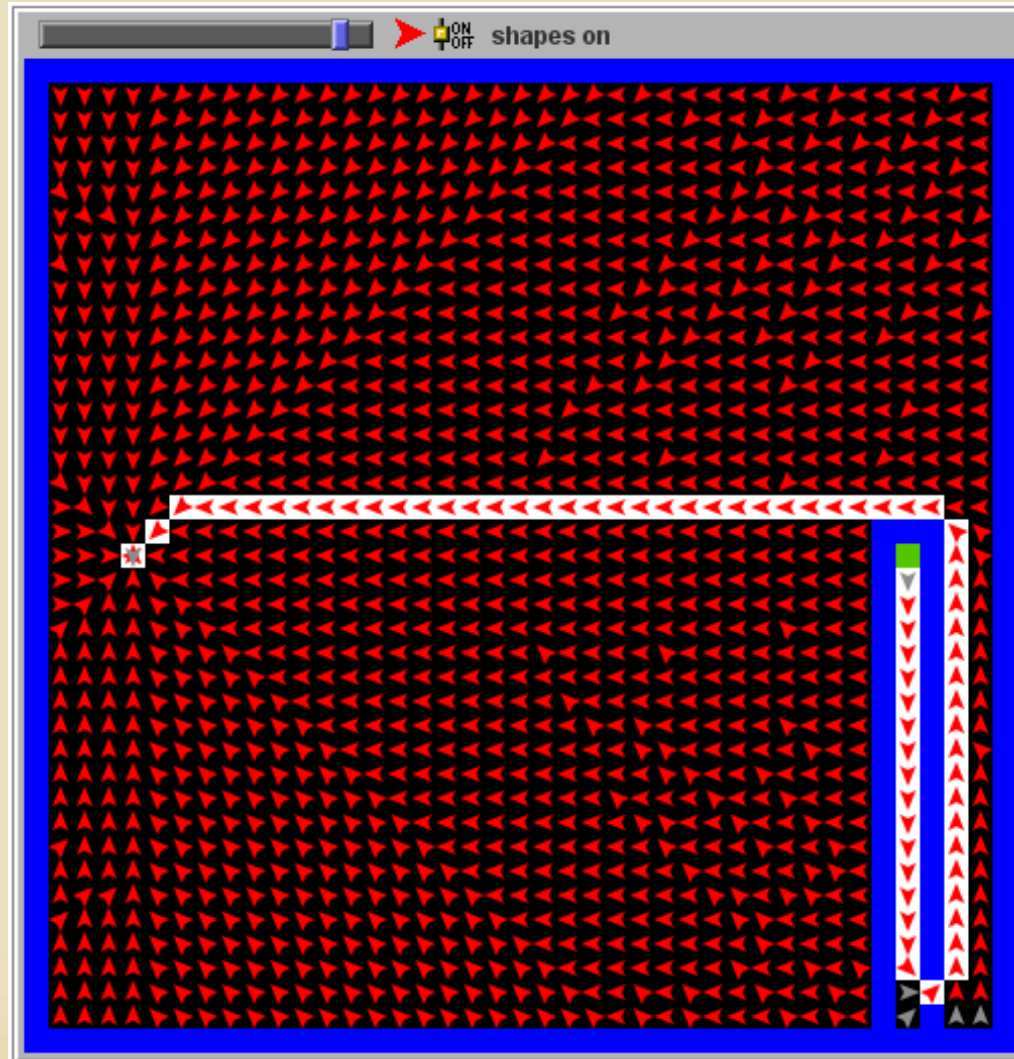


# A\*-Applet



# Lösungen zu Applet-Aufgaben

- Aufgabe 2)



# Lösungen zu Applet-Aufgaben

- Aufgabe 3)

