# A\*

Der A\* Algorithmus gehört zu den informierten Suchverfahren, was bedeutet, dass er über Wissen bezüglich des Suchraums verfügt, welches genutzt wird um die Suchzeit zu verringern.

Der Algorithmus verwendet sowohl die realen Pfadkosten als auch die Schätzung der Restkosten (Heuristik) eines Knotens. Die Kostenfunktion lautet dabei wie folgt: f(n) = g(n) + h(n)

Es wird immer ein Weg gefunden (sofern vorhanden) und dieser ist auch immer der optimale (bestmögliche) Weg

Als Datenstruktur zur Speicherung der Pfade wird eine sortierte Queue (Prioritätsqueue genutzt)

## Regeln für die Heuristik:

Damit die Optimalität und die Vollständigkeit des A\* Algorithmus gewährleistet werden können, müssen einige Kriterien bezüglich der Heuristik erfüllt sein.

So ist die Heuristik z.B. nur dann zulässig, wenn für alle Knoten gilt, dass die Schätzung der Restkosten (h(n)) kleiner/gleich den tatsächlichen optimalen Restkosten eines Knotens ist.

Außerdem muss die Heuristik für jeden Knoten größer/gleich 0 sein und für den Zielknoten explizit 0 betragen.

## Vorgehensweise

Die Vorgehensweise des A\* Algorithmus wird im Folgenden Stichpunktartig beschrieben:

* Schritt 1: Nimm den Startknoten, berechne die Gesamtkosten (bestehend aus realen Pfadkosten und geschätzten Restkosten) für den Knoten und füge ihn in eine sortierte Queue ein.
* Schritt 2: Nimm den ersten Knoten aus der Queue, kalkuliere die Gesamtkosten für alle Nachbarn und füge die Pfade in die Queue ein
* Schritt 3: Lösche alle Pfade aus der Queue, deren Gesamtkosten größer oder gleich denen eines anderen Knotens in der Queue sind, sofern beide Pfade auf den gleichen Knoten verweisen
* Schritt 4: Sortiere die Queue, ausgehend von den Gesamtkosten der Pfade
* Schritt 5: Überprüfe ob der erste Knoten in der nun sortierten Queue auf den Zielknoten verweist
* Wenn in Schritt 5 festgestellt wird, dass der Zielknoten gefunden wurde ist der Algorithmus fertig. Andernfalls werden die Schritte 2-5 solange durchlaufen, bis der erste Knoten in der sortierten Queue auf den Zielknoten verweist oder bis die Queue leer ist.

# Branch and Bound

Der Branch and Bound Algorithmus gehört ebenfalls zu den informierten Suchverfahren.

Im Gegensatz zu dem A\* Algorithmus verwendet der Branch and Bound Algorithmus ausschließlich die realen Pfadkosten, weswegen die Kostenfunktion wie folgt lautet: f(n) = g(n)

Es wird immer ein Weg gefunden (sofern vorhanden) und dieser ist auch immer der optimale (bestmögliche) Weg

Als Datenstruktur zur Speicherung der Pfade wird eine sortierte Queue (Prioritätsqueue genutzt)

## Vorgehensweise

Die Vorgehensweise des Branch and Bound Algorithmus ist die gleiche wie bei dem A\* Algorithmus mit dem Unterschied, dass die Berechnung der Gesamtkosten lediglich aus den realen Pfadkosten und nicht aus der Summe der realen Pfadkosten und der Heuristik besteht.

# Best First

Genauso wie der A\* und der Branch and Bound Algorithmus, gehört auch der Best First Algorithmus zu den informierten Suchverfahren

Der Best First Algorithmus setzt lediglich auf die Heuristik, was bedeutet, dass die realen Pfadkosten hier überhaupt keine Rolle spielen.

Im Gegensatz zu den A\* und Branch and Bound Algorithmen ist der Best First Algorithmus weder optimal, noch vollständig.

Als Datenstruktur zur Speicherung der Pfade wird ebenfalls eine sortierte Queue (Prioritätsqueue genutzt)

## Vorgehensweise

Auch der Best First Algorithmus verfolgt die gleiche Strategie wie der Branch and Bound und der A\* Algorithmus, mit dem Unterschied, dass hier lediglich auf die Heuristik (die Schätzung der Restkosten vom aktuellen Knoten zum Zielknoten) gesetzt wird und die realen Pfadkosten keine Rolle spielen.