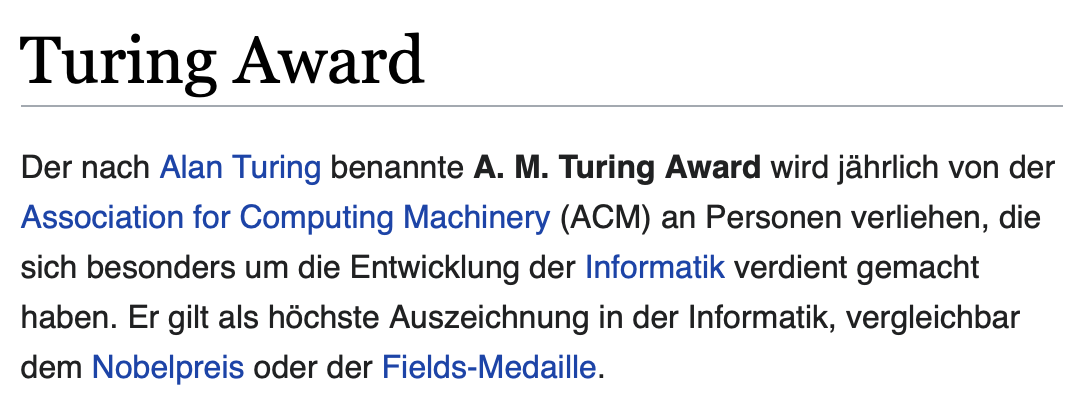


<https://de.wikipedia.org/wiki/Edsger_W._Dijkstra>



<https://de.wikipedia.org/wiki/Turing_Award>

Ein Bild, das Diagramm enthält.

Automatisch generierte Beschreibung

<https://de.wikipedia.org/wiki/Dijkstra-Algorithmus>

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

<https://de.wikipedia.org/wiki/Dijkstra-Algorithmus#Anwendungen>

# Der Algorithmus

## Schritt 1: Startaufstellung

Markiere im Graphen zunächst den Startknoten als besucht (grün). Trage die Distanz zum Starknoten mit 0 ein. Dieser Startknoten wird zum aktuellen Knoten.

## Schritt 2: Distanzen zu den Nachbarknoten

Für alle Nachbarknoten des aktuellen Knotens, die noch nicht als besucht markiert sind (also noch nicht grün sind), erledigst du folgende Punkte:

* Berechne die Entfernung zum Startknoten: Entfernung zum aktuellen Knoten + Entfernung zum Nachbarn.
* Wenn der betroffene Nachbar noch keine Entfernung eingetragen hat, dann trage die gerade berechnete Entfernung einfach ein und merke dir den aktuellen Knoten als Vorgänger des Nachbarn.
* Wenn der betroffene Nachbar bereits eine Entfernung eingetragen hat und diese größer ist, als die gerade berechnete Entfernung, dann trage die neue Entfernung ein und merke dir den aktuellen Knoten als Vorgänger des Nachbarn.

## Schritt 3: Weiterreise

* Wenn es nicht besuchte Knoten gibt, suchst du denjenigen mit der aktuell kleinsten Entfernung aus. Markiere diesen Knoten als besucht. Dieser Knoten wird der aktuelle Knoten. Fahre mit Schritt 2 fort.
* Wenn es keine nicht besuchten Knoten mehr gibt (alle Knoten sind grün), dann ist der Algorithmus zu Ende. Du hast nun die kürzesten Wege vom Startknoten aus zu allen restlichen Knoten des Graphen berechnet.

## Der kürzeste Weg für alle zum Startknoten

Wenn du nun den kürzesten Weg von einem beliebigen Knoten des Graphen zum gewählten Starknoten wissen möchtest, dann kannst du von diesem Knoten aus über die jeweiligen Vorgänger den kürzesten Weg zum Startknoten finden.