



## PROLOG

---

# Reisplanner

---

February 19, 2021

*Student:*  
René Kok  
13671146

*Lecturer:*  
Koen van Elsen

*Course:*  
Programmeertalen

*Course code:*  
5062PROG6Y

## 1 Introductie

Voor de derde week van programmeertalen is het de opdracht om een reisplanner te programmeren voor de dienstregeling van de Nederlandse Spoorwegen.

In dit document wordt er antwoord gegeven op de vragen gesteld in de opdracht.

## 2 Paden zoeken (Ontwikkeld)

**Welke operator wordt er gebruikt voor het vergelijken van de naburige knoop met de knoop gegeven door To en waarom?**

Voor het vergelijken van de naburige knoop met de knoop gegeven door To wordt de "==" operator gebruikt. Deze operator is bedoeld om te controleren of een variabele gelijkwaardig is aan een andere variabele in prolog.

**Waarom mag de betreffende naburige knoop niet onderdeel zijn van een kant die aanwezig is in Visited, de lijst van kanten die al bezocht zijn?**

Wanneer de betreffende naburige knoop onderdeel is van een kant die aanwezig is in Visited, zijn er een oneindig aantal routes mogelijk omdat de knopen eindeloos herhaald mogen worden.

**Welke paden zijn er van 1 naar 3, van 3 naar 5 en van 5 naar 4?**

1 to 3: [edge(1,2,5), edge(2,3,4)]  
3 to 5: [edge(3,1,9), edge(1,2,5), edge(2,5,5)]  
3 to 5: [edge(3,2,2), edge(2,5,5)]  
5 to 4: [edge(5,1,3), edge(1,2,5), edge(2,4,3)]  
5 to 4: [edge(5,4,2)]

### 3 Het kortste pad vinden (Competent)

**Wat zijn de kosten van ieder pad van 5 naar 4 om deze te bewandelen?**

Path : [edge(5,1,3), edge(1,2,5), edge(2,4,3)]

Cost : 11

**Wat zijn de kortste paden van 1 naar 3, van 3 naar 5 en van 5 naar 4?**

1 to 3 : [edge(1,2,5), edge(2,3,4)]

3 to 5 : [edge(3,2,2), edge(2,5,5)]

5 to 4 : [edge(5,4,2)]