

# Grundlagen der Wissensverarbeitung – Tutorial 3, Gruppe 4

Arne Beer MN 6489196      Marta Nevermann MN 6419716  
Daniel Waller MN 6813853

## Exercise 1.3

### 1. - 3.

siehe Code

### 4.

Die Breitensuche findet zwar den kuerzesten Weg, arbeitet allerdings auch einen sehr großen Teil des Grids ab und ist dementsprechend inperformant.

Die Laufzeit der Tiefensuche ist im optimalen Fall der kuerzeste Weg, im schlechtesten Fall, wird das komplette Grid abgelaufen. Bei der Tiefensuche ist auch nicht garantiert, dass der schnellste Weg gefunden wird.

Die Probleme liessen sich beide durch eine Gewichtung umgehen, der Gewichtung zwischen zwei Teilen des Grids. Dadurch wäre die Suche ähnlich einer priorisierten Breitensuche.

### 5.

In beiden behandelten Fällen um eine blinde Suche, da ohne Heuristiken zur Zielfindung gearbeitet wird. Im Zweifelsfall ist diese Art der Suche ineffizienter, da die Suche ohne weitere Information zur Zielfindung “blind” den Baum abarbeitet. Bei der Tiefensuche könnte etwa das Problem auftauchen, dass die Suche aufgrund eines unendlichen Zweigs oder eines Zyklus niemals stoppt. Die Breitensuche ist sehr speicheraufwändig (der Speicheraufwand wächst exponentiell), was bei großen Suchräumen und/oder wenig vorhandenem Speicher zu Problemen führen kann. Diese Probleme würden z. Bsp. bei einer Applikation zur Navigation oder einem Routenplaner für den ÖPNV entstehen.

## 6.

Durch sogenannte Heuristiken werden Informationen über das Ziel dazu verwendet, die Suche zu verkürzen bzw. die Suche effizienter zu gestalten. Eine Heuristik kann man durch das Lösen einer Vereinfachung des Problems finden. Um zu vermeiden, dass die Suche in einem Zyklus hängen bleibt, entfernt man einen Pfad, der zu einem bereits erreichten Knoten führt, ohne dabei die bestmögliche Lösung zu entfernen.