

GWV-Abgabe zum 31.10.2014

Arne Struck, Knut Götz

31. Oktober 2014

1.1

1.

Repräsentation: siehe java-files.

2.

BFS-Impl: siehe java-files.

Beispieloutput (eigentlich keine linebreaks, die sind nur für verbatim):

Starting BFS.

Goal found!

The path: START EAST EAST EAST NORTH NORTH NORTH EAST EAST EAST EAST SOUTH
EAST EAST EAST EAST SOUTH SOUTH WEST WEST WEST SOUTH SOUTH SOUTH EAST EAST

```
xxxxxxxxxxxxxxxxxxxxx
x      ppppp      x
x      pxxppppp   x
x      px xxxxp   x
x  sppp  x pppp   x
x      x x pxxxxxx
x  xx xxxxx p     x
x      x   ppg    x
x      x         x
xxxxxxxxxxxxxxxxxxxxx
```

3.

DFS-Impl: siehe java-files.

Beispieloutput (eigentlich keine linebreaks, die sind nur für verbatim):

Starting DFS

Goal found!

The path: START NORTH NORTH NORTH EAST EAST EAST EAST EAST EAST EAST
EAST EAST EAST EAST EAST EAST SOUTH SOUTH WEST WEST WEST SOUTH WEST WEST
WEST WEST SOUTH SOUTH EAST EAST EAST SOUTH

```

xxxxxxxxxxxxxxxxxxxxx
x  ppppppppppppppppx
x  p   xxx       px
x  p   x xxxxxppppx
x  s       xppppp  x
x        x xp xxxxxxx
x  xx xxxxxpppp   x
x        x       g   x
x        x       x   x
xxxxxxxxxxxxxxxxxxxxx

```

4.

- DFS findet nicht den kürzesten Pfad.
- DFS besucht potentiell weniger Knoten, als BFS (best case)
- BFS kann unter Umständen eine größere Frontier aufbauen

5.

Geht man davon aus, dass es in dem Maze ein Ziel gibt, dass das Ziel vom Start zu erreichen ist, d.h. es gibt einen Pfad und dass das Maze nicht extrem groß ist (sodass es bei BFS zu Speicherproblemen führt), dann sehen wir keine Probleme. Weitere Bedingungen für unsere Implementation sind, dass genau ein Start und ein Ziel definiert ist. Beide Suchverfahren werden (früher oder später) einen Pfad finden.

Allerdings sind die Suchverfahren für bestimmte Environments besser bzw. schlechter geeignet.

```

xxxxxxxxx
xg      x
x      x
x      x
x      x
x      x
x      sx
xxxxxxxxx

```

Ein BFS besucht in diesem Fall alle Punkte während eine DFS, die zuerst nach Westen und (wenn Westen blockiert) dann nach Norden sucht, wesentlich schneller das Ziel erreicht.

Ein Maze, dass BFS bevorzugt wäre schwieriger (nur mit Wissen über die Reihenfolge in der der spezifische DFS auf den Stack pusht) zu konstruieren, dies ist allerdings bei weitem nicht unmöglich.

6.

Man kann DFS so erweitern, dass auch der kürzeste Pfad gefunden wird. Dann müssen alle möglichen Pfade betrachtet und daraufhin in ihrer Länge verglichen werden. Damit treibt man jedoch die Raumkomplexität in die Höhe.