

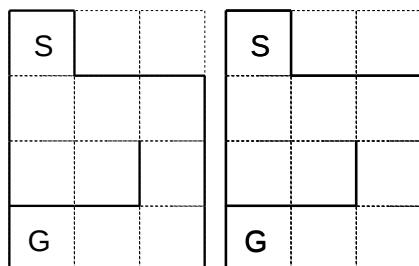
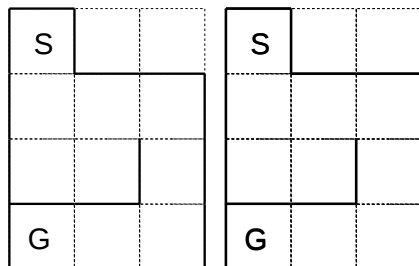
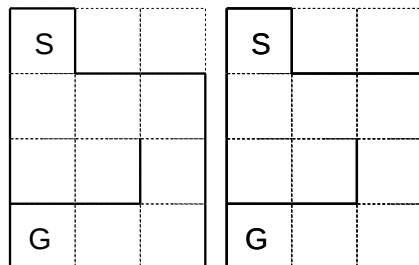
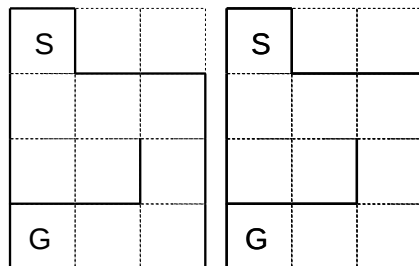
GWV – Grundlagen der Wissensverarbeitung

Tutorial 4 : Searching

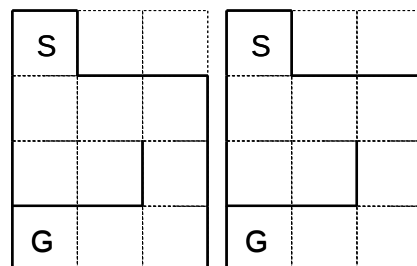
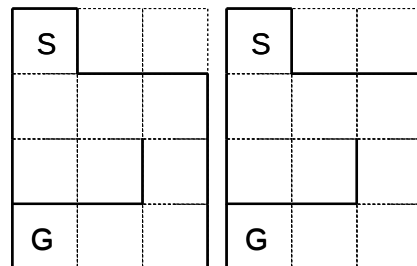
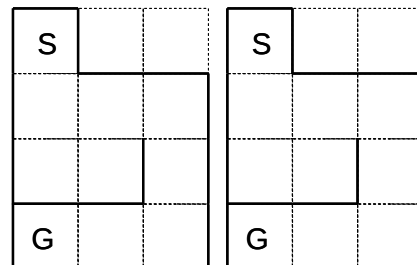
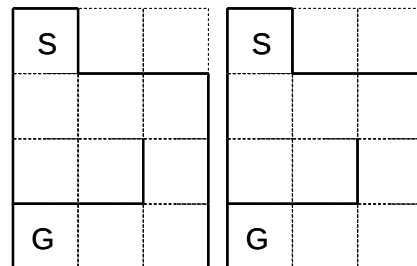
Class Exercise 4.1 : (Frontiers and Paths)

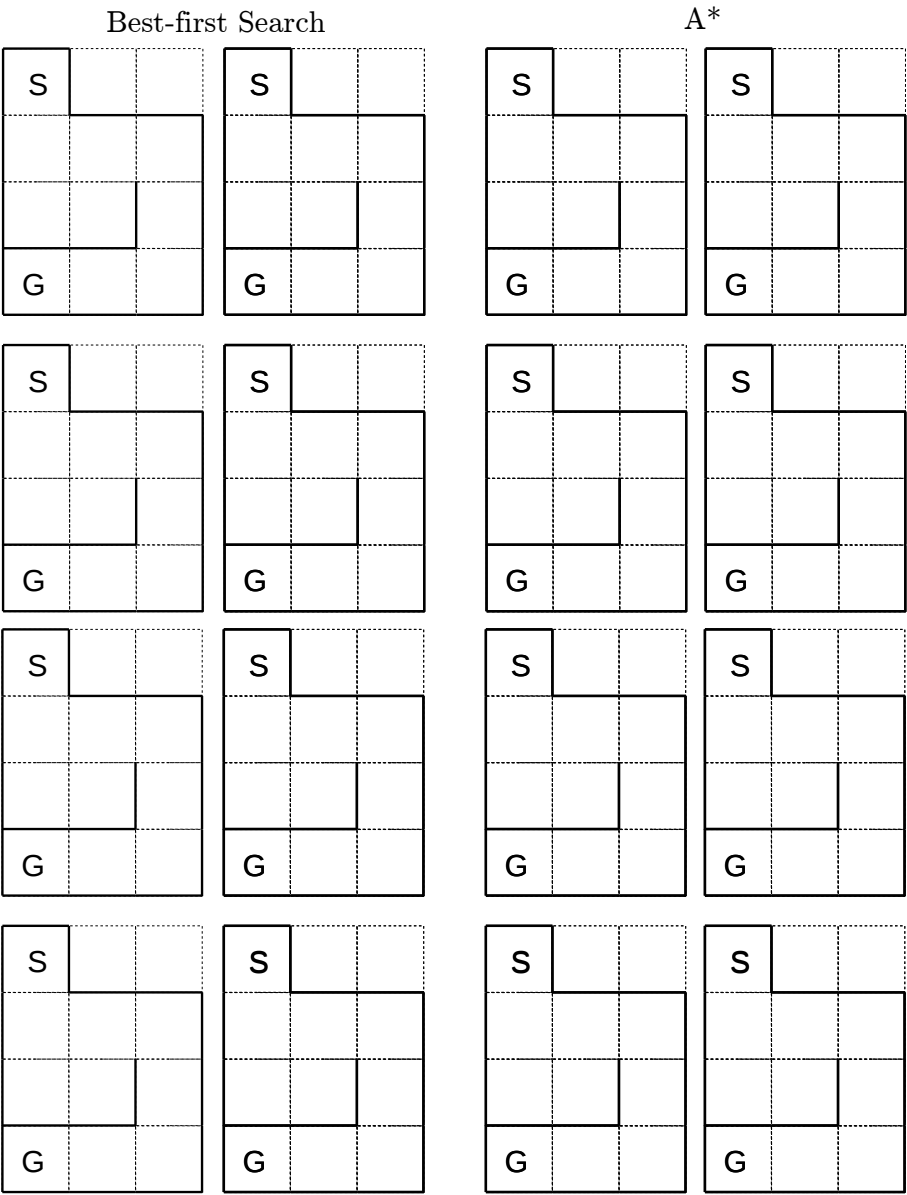
The grid below contains a small maze with a start and a goal. Suppose you use *Depth-first search* (different variants), *Best-first search* and A^* to find a path to the goal. For each search strategy, draw the frontier and the paths found so far in each search step. You have space for at most 8 search steps per search strategy. For *Depth-first search*, perform the search for two variants, the first one tries to search *up* first, then *right* etc. The other one starts with *down*, then *left* etc.

Depth-first Search
Priorities: (up, right, down, left)



Depth-first Search
Priorities: (down, left, up, right)





Exercise 4.2: (Blind Search)

```

XXXXXXXXXXXXXXXXXXXXX
X                    X
X          xxx       X
X          X xxxxxx  X
X    s      X        X
X          x x       X
X          x x       X
X  xx xxxxxx        X
X          x         X
X          x         X
X          x         X
XXXXXXXXXXXXXXXXXXXXX

```

Figure 1 shows the known environment for a robot in an ASCII-Art representation. Again the robot starts in the field **s** (start) and wants to get to the field **g** (goal). The robot can move one field at a time in any of the four directions (up, down, left, right). The fields with an **x** denote a blocked field that the robot can not enter. *Hint: Again you will find the text files specifying the environments in the nats wiki.*

Extend the program from last week to perform search in the maze:

1. Implement the blind search strategy “breadth first” to find a path for the robot. (3 Pt.)
2. Implement the blind search strategy “depth first” to find a path for the robot. (3 Pt.)
3. Provide example mazes that show the differences between and properties of the search strategies. Describe these properties. (2 Pt.)
4. Are there cases in which your program is unable to find a solution? Provide example mazes. (1 Pt.)

```

XXXXXXXXXXXXXXXXXXXXX
X2                    X
X          xxx       X
X          X xxxxxx  X
X    s      X        X
X          x x       X
X          x x       X
X  xx xxxxxx        X
X          x         X
X          x         X
X          x         X
X    1  x  2         X
XXXXXXXXXXXXXXXXXXXXX

```

5. The Figure above shows a modified environment. There are mysterious portals that transport the robot from one field to a pre-determined field far away. They are denoted with numbers, that is if the robot enters a field with a 1, it is automatically teleported to the other field with a 1 and so on. Modify your program to cope with this change. (3 Pt.)

Version: November 7, 2019
Achievable score on this sheet: 12

of
12