TDT4136 - Exercise 3

Stein-Otto Svorstøl and Andreas Drivenes
3rd year MTDT

Fall 2014

We based our A*-implementation on the work done by Luarent Luce, see the bibliography. [1] The visualization is simply written as text to console, as we did not make visualization a priority.

We've structured our code so that there's one Python-file for each subtask, except the third one where each implementation has gotten it's own file.

Subproblem A-1

Here are the results from the four boards:

Board 1-1

```
....................
........ooooooooo..
........o######..o..
........oooA..#..B..
.........######.....
....................
....................
```

Board 1-2

```
....ooo#............
...oo#o#............
..oo#oo#............
Aoo#.o#....oooooooooB
....#oo#..oo#.......
.....#oo#oo#........
......#ooo#.........
```

Board 1-3

```
.........oooooo.....
.........o#...ooo...
.......##oo#....o...
......#oA#o#....o...
......#o#oo#....o...
......#ooo#.....oo..
.......###........ooB
```

Board 1-4

```
Ao#.......#......#..
#o#.#####.#.####.#..
oo#ooooo#.#....#....
o##o###o######.#####
oo#oB#oo#....#...#..
#o####o##.##.#.#.##.
.oooooo....#...#....
```

Subproblem A-2

Board 2-1

```
mmmmmfffffrrrrrrrrrArrrrrrrrrrrrrrrfffmmmmm
mmmfffffffffrrrrrrooooooooooooooorfffffmmmm
mmfffffffffffffffffffffffffffffofffffmmmm
mmfffffffffffffwwwwwfffffffffofffffffmmm
mfffffffffffffwwwwwwwfffffffffofffffmmmm
mmfffffffffffffwwwwwwwffooooooorrrrrrmmmm
mmmfffffffffffffwwwwwfffofffffffffrffffmmm
mmfffffffffffffffffffffofffffffffrffffmm
mmfffffffffggggggggggggogggggggggggffffmm
mmmffffggggggggggBooooooggggggggggggffmm
```

Board 2-2

```
fffffffffffgggrgggggggrggggfffffffrrffffff
ffAffffffgggrrgggggggrrggfffffffffrrffffff
ffofffgggggrrgggggggrrgggffffrrrrrrffffffff
ggogggggggggrggggrrrrgggffffrrffffffffffff
ggooooooooooooooogggggffffrrffffffffffffff
ggggrrgggggrggggogggffffffrffffffffffffff
gggrrgggggggrrgggoggfffooooooofffffffffff
ggrrgggffgggrrrrogffooorfrfforrrrffffffff
ggrggfffffffffffrooooofffffrffofffrrffffff
ggrgfffffffffffffffffffffrffBffffrrffffff
```

Board 2-3

```
ggggggggwwwggggmmmmmmmmmBrrrrrrrmmmmm
ggggggggwwwwggggmmmmmmmmmmommmmmrmgggg
gggggggggwwwwggggmmmmmmmmmommmmggrggggg
ffgggggggggwwwwggggmmmoooooogggggrrgggg
ffgggggggggwwwwwwwwwwowwgggggggggrrrrr
ffffgggggggggwwwwwwwwowwwwgggggggggggg
ffooooooooooooooooooowwwwwwgggggmmmmm
fAoffffffffgggggggmmmmmmwwwwmmmmmmmmm
fffffffffffffmmmmmmmmmmmmmwwwmmmmmmmm
ffffffffffmmmmmmmmmmmmmmmmwwwmmmmmmmm
```

Board 2-4

```
wwwwwgggggggggggggoooooooooooooooorrrrr
wwwwwwwgggggggggoooogggggggwwwwwgorgggg
wwwwwwwwwwgggAoogggggwwwwwwwwwwwowgggg
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwowwwww
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwowwwww
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwoowwww
wwwwwwwwgggggBgggggwwwwwwwwwwwwowwwwww
wwwwgggggfffoffgggggggggwwwwwwwwowwwww
wwgggfffffffoffffffgggggggggggrogwwwww
wgggfffrrrrrrooooooooooooooooooogggggg
```

Subproblem A-3

We've used the same syntax as the figures given to us in the problem description, but as we did not prioritize the visualization part of the exercise, we were not able to hand in a layered presentation. This means we cannot show the type of cell.

In general we can note that Djikstra uses a very long time to calculate it's solution, especially on board 2-3.

We can see that A* finds the shortest path easily, while Djikstra spends more time and has to go further. BFS "explores" everything until it finds it's goal. This c

Board 1-1

A*

```
........**********..
.......*oooooooooo*.
......*xo######*xo*.
.....*xxoooAxx#xxB..
......*xx######xx*..
.......*xxxxxxxx*...
........********....
```

Djikstra

```
xxxxxxxxxxxxxxxxxx*
xxxxxxxxoooooooooxx*
xxxxxxxxo######xoxx*
xxxxxxxxooAxx#xoB*.
xxxxxxxxx######xxxx*
xxxxxxxxxxxxxxxxxx*
xxxxxxxxxxxxxxxxx*.
```

BFS

```
ooo*ooo*ooo*ooo*....
o*o*o*ooo*ooo*oo*...
o*o*o*xx*######o*...
o*o*o*x*ooA*.#o*B..
o*ooo*x*o######o*o*.
o*******o*xxxx*o*o*.
ooooooooo*xxxxxooo*.
```

Observations

We see that both Djikstra and A* finds the shortest path, but Djikstra explores nearly the entire graph (a lot of closed nodes). BFS finds a path, but the search is quite "stupid", as it goes up and down many times. A* has the most open nodes.

Board 1-2

A*

```
xxxxooo#............
xxxoo#o#............
xxoo#oo#.**********.
Aoo#xo#.*xxooooooooB
xxxx#oo#*xoo#******.
xxxxx#oo#oo#........
xxxxxx#ooo#.........
```

Djikstra

```
xooooo#xxxxxxxxx*..
xoxxx#o#xxxxxxxxx*.
ooxx#oo#xxxxxxxxxx*
Axx#xo#xxxxooooooooB
xxxx#oo#xxoo#xxxxxx*
xxxxx#oo#oo#xxxxxx*.
xxxxxx#ooo#xxxxxx*..
```

BFS

```
xooooo#ooo*ooo*ooo*
xo*xx#o#o*o*o*o*o*o*
*oo*#oo#o*o*o*o*o*o*
A*o#*o#xo*ooo*o*o*oB
o*o*#oo#oo*x#*o*o*xx
o*o*.#oo#o*#x*o*o*xx
ooo*..#ooo#xxxooo*xx
```

Observations

A* and Djikstra finds the same path, but Djisktra closes more node (see the right hand side of the board.) BFS finds a path, but explores a lot.

Board 1-3

A*

```
........*oooooox*...
.......*xo#xxxooo**.
.......##oo#xxx*oxx*
......#oA#o#*xxxo*xx
......#o#oo#*xxxoxxx
......#ooo#.*xxxooxx
.......###...*xx*ooB
```

Djikstra

```
xxxxxxxxxooooxxxxxxx
xxxxxxxxxo#xooxxxxxx
xxxxxxx##oo#xooxxxxx
xxxxxx#oA#o#xxoxxxxx
xxxxxx#o#oo#xxoxxxxx
xxxxxx#ooo#xxxoooxxx
xxxxxxx###xxxxxxooB%
```

BFS

```
xxxxxxxxxooo*xooo*xx
xxxxxxxxxo#oo*o*o*xx
xxxxxxx##oo#o*o*o*xx
xxxxxx#oA#o#o*o*o*xx
xxxxxx#o#oo#o*o*o*xx
xxxxxx#ooo#*o*o*o*xx
xxxxxxx###xxooo*oooB
```

Observations

Again we see that Djikstra closes more nodes than A*, but this time it has a slightly different path - it goes downward before A* does. Djikstra has closed all the nodes on the left hand side of the board, A* on the other hand moved towards the right hand side at once. BFS also closed the whole left part of the board, then moves up an down on the right side. It has the most closed nodes, and the longest path to the goal.

Board 1-4

A*

```
Ao#.......#......#..
#o#*#####.#.####.#..
oo#ooooo#.#....#....
o##o###o######.#####
oo#oB#oo#xx*.#...#..
#o####o##x##.#.#.##.
xooooooxxxx#...#....
```

Djikstra

```
Ao#xx*....#......#..
#o#x#####.#.####.#..
oo#ooooo#.#....#....
o##o###o######.#####
oo#oB#oo#xxxx#...#..
#o####o##x##x#*#.##.
xooooooxxxx#xxx#....
```

BFS

```
Ao#.......#......#..
#o#*#####.#.####.#..
oo#ooooo#.#....#....
o##o###o######.#####
```

```
oo#oB#oo#....#...#..
#o####o##.##.#.#.##.
xooooooo*...#...#....
```

Observations

Here all the three algorithms almost has the same path, but this is not a board where you can find that many possible paths to the goal. Djikstra and A* has about the same amount of open nodes, although A* has a little bit more. BFS has not that many closed nodes this time, actually fewer than both A* and Djikstra. It seems this board fits BFS the most.

Board 2-1

A*

```
mmm*xxxxxxxxxxxxAxxxxxxxxxxxxxxxxxx*mmm
mmmf*xxxxxxxxxxxxooooooooooooooxxxxx*mmmm
mmff*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*mmm
mmfff*xxxxxxxxxx*x*xxxxxxxxxxoxxxxxxx*mm
mffffff*xxxxxxxx*w*w*xxxxxxxxxxoxxxxxx*mm
mmffffff*xxxxxxx*www*xxxoooooooxxxxxxx*mm
mmmffffff*xxxxx*wwww**xxoxxxxxxxxxxxxx*m
mmfffffff*xxxxx*ff*xxxxoxxxxxxxxxxxxx*m
mmffffffff*xxxxx*g*xxxxoxxxxxxxxxxxx*mm
mmmffffgggg**x**gBooooooxxxxxxxxxxxx*mm
```

Djikstra

```
mm*xxxxxxxxxxxxooAxxxxxxxxxxxxxxxxxx*mm
m*xxxxxxxxxxxooooxxxxxxxxxxxxxxxxxxx*mm
mm*xxxxxxxxxxoxxxxxxxxxxxxxxxxxxxxx*mm
mmf*xxxxxxxxxoxxxxxxxxxxxxxxxxxxxxx*m
mfff*x*xxxxxxox*****xxxxxxxxxxxxxxxx*m
mmfff**xxxxxxox*www*xxxxxxxxxxxxxxxx*m
mmmffff*xxxxxoxx***xxxxxxxxxxxxxxxxx*m
mmffffff*xxxxooxxx*xxxxxxxxxxxxxxxxx*m
mmffffff**xxxxooxxxxxxxxxxxxxxxxxxx*mm
mmmffff*xxxxxxxooB*xxxxxxxxxxxxxxxxx*mm
```

BFS

```
mmmmmffffrrrrrrr*A*rrrrrrrrrrrrfffmmmmm
mmmfffffffffrrrrr*o*rrrrrrrrrrrrfffffmmmm
mmfffffffffffffff*o*ffffffffffrffffffmmmm
mmffffffffffffw*o*wffffffffffrffffffmmm
mfffffffffffww*o*wwffffffffrffffffmmmm
```

```
mmfffffffffffww*o*wwffrrrrrrrrrrrrrmmmm
mmmfffffffffffw*o*wfffffffffffrffffmmm
mmffffffffffffff*o*fffffffffffffrffffmm
mmfffffffggggggg*o*gggggggggggggggffffmm
mmmffffggggggggggBgggggggggggggggggggffmm
```

Observations

First of all we note that A* and Djikstra has found two different paths, and that A* has fewer closed nodes. A* utilizes the road for maximal speed around the water, whereas Djikstra does not. BFS does not care about anything, and swims across the water. This result makes us question our heuristic function, as the A*- Djikstra-solution differ so much from each other.

Board 2-2

A*

```
xxxxxxxxxxxxxxxxxxxxxxxxxx*ffffffrffffff
xxAxxxxxxxxxxxxxxxxxxxxxx*ffffffrffffff
xxoxxxxxxxxxxxxxxxxxxxxxx*frrrrrffffffff
xxoxxxxxxxxxxxxxxxxxxxxx*x*rffffffffffff
xxooooooooooooooxxxxxxx**x**fffffffffff
xxxxxxxxxxxxxxxoxxxxxx*xxxxx*f*ffffffff
xxxxxxxxxxxxxxxoxxxxxooooooox*x*fffffff
xxxxxxxxxxxxxxxoxxxooooxx*xxoxxxx*ffffff
xxxxxxxxxxxxxxxooooooxxx*r*xo*x*x*ffffff
xxxxxxxxxxxxxxxxxxxxxx*frf*Bf**x*rfffff
```

Djikstra

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxx****rffffff
xxAxxxxxxxxxxxxxxxxxxxxxxxxxxxx*ffffff
xxooooxxxxxxxxxxxxxxxxxxxxxxxxxx*fffff
xxxxooxxxxxxxxxxxxxxxxxxxxxxxxx*ffffff
xxxxxooooooooooooxxxxxxxxxxxxx**fffffff
xxxxxxxxxxxxxxxxoxxxxxxxxxxxxxx*ffffff
xxxxxxxxxxxxxxxxoxxxxxxoooooxxxxxx*fffff
xxxxxxxxxxxxxxxxoxxxxxxooxxooxxxxxx*ffff
xxxxxxxxxxxxxxxxoooooooxxxxoxxxxxx*fffff
xxxxxxxxxxxxxxxxxxxxxxxxxxoB*****rfffff
```

BFS

```
xx*xooo*ooo*ooo*ooo*ooo*ooo*fffffrrffffff
x*A*o*o*o*o*o*o*o*o*o*o*o*o*ffffrrffffff
```

```
x*o*o*o*o*o*o*o*o*o*o*o*o*o*rrrrrfffffff
x*o*o*o*o*o*o*o*o*o*o*o*o*o*rfffffffffff
x*o*o*o*o*o*o*o*o*o*o*o*o*o*ffffffffffff
x*o*o*o*o*o*o*o*o*o*o*o*o*o*ffffffffffff
x*o*o*o*o*o*o*o*o*o*o*o*o*o*rfffffffffff
x*o*o*o*o*o*o*o*o*o*o*o*o*o*rrrrrfffffff
x*o*o*o*o*o*o*o*o*o*o*o*o*o*ffffrrffffff
xxooo*ooo*ooo*ooo*ooo*ooo*ooBfffffrffffff
```

Observations

We note that BFS has a lot of closed nodes as usual, and walks up and down - not very surprising, as it searches in breadth first. Seems A* and Djikstra almost has the same amount of closed nodes, and has almost the same path. Our heuristic function fits this board well it seems.

Board 2-3

A*

```
xxxxxxxxxxxxxxxxxxxxxxxxxBr*r*xxxx*x*x
xxxxxxxxxxxxxxxxxxxxxxxxxo*x*xxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxoxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxooooooxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxoxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxoxxxxxxxxxxxxxxx
xxooooooooooooooooooooxxxxxxxxxxxxxx*
xAoxxxxxxxxxxxxxxxxxxxxxxxxxxxxx****m
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx***mmmmm
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx**mmmmmmm
```

Djikstra

```
xxxxxxxxxxxxxxxxxxxxxxx*Box*rr*xxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxox***xxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxoxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxoxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxooxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxooooxxxxxxxxxxxx
xxxxooooooooooooooooooxxxxxxxxxxxxxx
xAoooxxxxxxxxxxxxxxxxxxxxxxxxxxxx****
xxxxxxxxxxxxxxxxxxxxxxxxxxxx*x*x*mmmm
xxxxxxxxxxxxxxxxxxxxxxxxx***m*m*mmmm
```

BFS

```
ooo*ooo*xxooo*xooo*ooo*ooo*Brrrrrrrmmmmm
o*o*o*oo*xo*oo*o*o*o*o*o*o*o*mmmmmrmgggg
o*o*oo*oo*oo*o*o*o*o*o*o*o*o*mmmggrggggg
o*oo*oo*oo*o*o*o*o*o*o*o*o*ggggggrrgggg
o**oo*oo*o*o*o*o*o*o*o*o*o*o*ggggggrrrrr
ooo*oo*o*o*o*o*o*o*o*o*o*o*o*ggggggggggg
x*oo*o*o*o*o*o*o*o*o*o*o*o*o*gggggmmmmm
*A*o*o*o*o*o*o*o*o*o*o*o*o*o*o*wmmmmmmmmm
*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*wwmmmmmmmm
xooo*ooo*ooo*ooo*ooo*ooo*ooo*ooo*wwmmmmmmm
```

Observations

Again we see that A* and Djikstra almost has the same path, and both closes almost every other node than the path. Nothing new about BFS here.

Board 2-4

A*

```
www*xxxxxxxxxxxxxxxoooooooooooooooxxxxx
wwww*xxxxxxxxxxxooooxxxxxxxxx*x*xxoxxxxx
wwwww*xxxxxxxxAooxxxxxxxxx*x**w*w*xoxxxxx
wwwwww**x*xxxxxxxxxxxxx**w*wwwww*xox*xx*
wwwwwwww*w*x*xx*x*x*x**wwwwwwwww**ox***w
wwwwwwwwwww*w***********wwwwww*xoox*www
wwwwwwwwg*g*gB*xxxxx*xxxx*****ww*ox*wwww
wwwwggg**x*x*oxxxxxxxxxxxxxxxx**xo**wwww
wwgggg*x*xxxxoxxxxxxxxxxxxxxxxxxxoxx*www
wgggf*xxxxxxxooooooooooooooooooox*gggg
```

Djikstra

```
w*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
ww*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
www*xxxxxxxxxxAxxxxxxxxxxxxxxxxxxxxxxx
wwww**xxxxxxxxoxxxxxxxxxx*xxx**x********
wwwwww******xxoxx********w*x*ww*wwrwwwww
wwwwwwwwwww*xxoxxxx*wwwwwww*wwwwwrrwwwww
wwwwwwwwgg*xoxoxxxxx*wwwwwwwwwwwwrwwwwww
wwwwggggg**xoooxxxxx*ggggwwwwwwwwrwwwwww
wwggggff*xxxxxxxxx**gggggggggggrrgwwwww
wgggfff*xxxxxxxxx*rrrrrrrrrrrrrrrgggggggg
```

BFS

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx*xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx*A*xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx*o*xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx*o*xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx*o*xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxBo*xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Observations

BFS has a very wrong solution to this one. Djikstra checks out a little more of the board than A* does (A* has less closed nodes).

General observations

We can see that A* finds the shortest path easily, while Djikstra spends more time and often has to go further. We know that Djikstra finds the shortest path when it returns it's result, but the cost is more calculations. A* performs very well on big boards where our heurstic function fits well. Eg. boards where the goal is straight to the left of A, A* checks out less nodes around the path, where Djikstra may check out all nodes around the path. We also see how A* uses it's heuristic function to maneuver in the right direction in cases where one starts in the middle of the board, and the goal is on the far right or far left. Djikstra first has to check out the other side of the board, before it can move in the right direction, whereas A* goes in the right direction at once. BFS "explores" everything until it finds it's goal, which is not always correct. We also note that when it comes to runtime, Djikstra is far slower than the two others, especially on the "board-2-3".
It all depends on our heuristic function, really.

References

[1] http://www.laurentluce.com/posts/solving-mazes-using-python-simple-recursivity-and-a-search/
Downloaded October 3. 2014