October 2014

Report, exercise 3 - A*-algorithm

Fredrik C. Berg and Håkon Ø. Løvdal

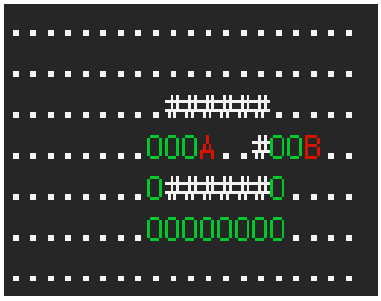
Subproblem A.1

A1.1

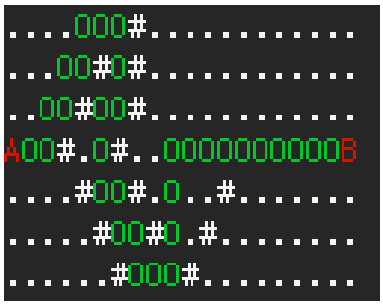
See “astar1.py” for commented source code

A1.2

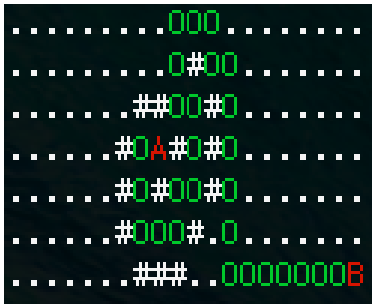
Board 1-1



Board 1-2



Board 1-3



Board 1-4

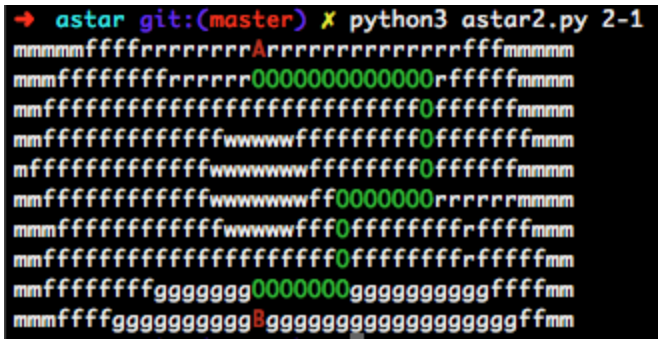
Subproblem A.2

A2.1

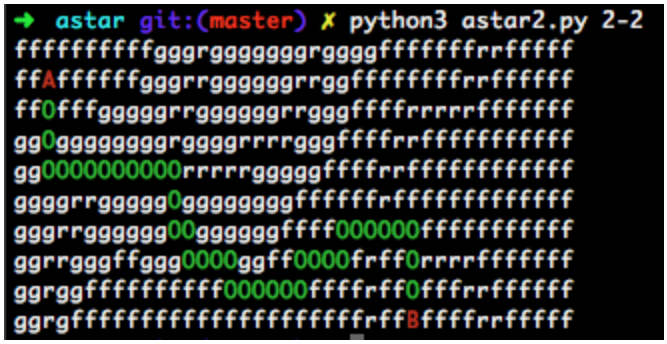
See “astar2.py” for commented source code

A2.2

Board 2-1



Board 2-2



Board 2-3

```
➜  astar git:(master) ✗ python3 astar2.py 2-3
ggggggggwwwggggmmmmmmmmmBOOOOOOOmmmmm
ggggggggwwwwggggmmmmmmmmmmmmmmmmmOmgggg
ggggggggggwwwwggggmmmmmmmmmmmmmmmmggOggggg
ffgggggggggwwwwggggmmmOOOOmmOOOOOOOrgggg
ffgggggggggwwwwwwwwwwOwwOOOOgggggrrrrr
ffffgggggggggwwwwwwwwOwwwwgggggggggggg
ffffffOOOOOOOOOOOOOOOOwwwwwwgggggmmmmm
fAOOOOOffffgggggggggmmmmmmwwwwmmmmmmmmmm
fffffffffffffffmmmmmmmmmmmmmmwwwmmmmmmmmm
ffffffffffmmmmmmmmmmmmmmmmmmmwwwmmmmmmmmm
```

Board 2-4

```
➜  astar git:(master) ✗ python3 astar2.py 2-4
wwwwwgggggggggggg0000000000000000grrrrr
wwwwwwwgggggggggggOgggggggggwwwwwOOrgggg
wwwwwwwwwwwgggAOOOOgggwwwwwwwwwwwwOwgggg
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwOwwwww
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwOwwwww
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwOOwwwww
wwwwwwwwggggg BggggwwwwwwwwwwwwwwOwwwwww
wwwwgggggggfffOffgggggggggwwwwwwwwOwwwwww
wwgggggfffffffOffffffgggggggggggOOgwwwww
wgggfffrrrrrr0000000000000000000000ggggggg
```

Subproblem A.3

A3.1

See the files “astar3_bfs.py” and “astar3_dijkstra.py” for commented source code.

A3.2

A*-algorithm:

Board 1-1

```
➜  astar git:(master) ✗ python3 astar1.py 1-1
....................
........*...........
.......*X######**...
......*XOOOAXX#OOB..
.......*O######O*...
.......*OOOOOOOO*...
........********....
```

Board 1-2

```
➜  astar git:(master) ✗ python3 astar1.py 1-2
...*OOO#............
..*OO#O#............
**OO#OO#.**********.
AOO#XO#.*OOOOOOOOOB
**XX#OO#*O**#******.
..*XX#OO#O*#........
...*XX#OOO#.........
```

Board 1-3

```
➜  astar git:(master) ✗ python3 astar1.py 1-3
........*OOO*.......
.......*XO#OO*......
.......##OO#O*......
......#OA#O#O*......
......#O#OO#O*......
......#OOO#*O******.
.......###.*OOOOOOOB
```

Board 1-4

```
➜  astar git:(master) ✗ python3 astar1.py 1-4
AO#.......#......#..
#O#*#####.#.####.#..
OO#OOOOO#.#....#....
O##O###O######.#####
OO#OB#OO#....#...#..
#O####O##.##.#.#.##.
*OOOOOOX*..#...#....
```

Board 2-1

```
➜  astar git:(master) python3 astar2.py 2-1
mmmm*XXXXXXXXXXXAXXXXXXXXXXXXXXXXX*mmmm
mmmf*XXXXXXXXXXXOOOOOOOOOOOOOOXXXXX*mmmm
mmfff*XXXXXXXXXXXXXXXXXXXXXXXXXOXXXXXX*mmm
mmffff*XXXXXXXX*****XXXXXXXXXOXXXXXXX*mm
mffffff*XXXXXX*wwwww*XXXXXXXXOXXXXXX*mmm
mmffffff*XXXXX*wwwww*XXOOOOOOOXXXXXX*mmm
mmmffffff*XXXXX*www*XXXOXXXXXXXXXXXX*mm
mmfffffffff*XXX*ff***XXXOXXXXXXXXXXXX*mm
mmfffffffffg*XX*g*OOOOOOOXXXXXXXXXXX*fmm
mmmffffggggg**gggB*XXXXXXXXXXXXXXXXX*mm
```

Board 2-2

```
➜  astar git:(master) python3 astar2.py 2-2
XXXXXX*fffg*XXX*ggg*XXX*ggfffffffrrffffff
XXAXXXX****XXXX****XXXX*ffff*fffrrffffff
XXOXXXXXXXXXXXXXXXXXX*fff*X*rrrfffffff
XXOXXXXXXXXXXXXXXXXX*fff*XX*fffffffffff
XXOOOOOOOOOXXXXXXXXX*fff*XX*fffffffffff
XXXXXXXXXXOXXXXXXXX**f***X**fffffffffff
XXXXXXXXXXOOXXXXX**XX*OOOOOO****fffffff
XXXXXXX**XXXOOOOXXXXOOOOXX**OXXXX*ffffff
XXXXX**ff***XXXOOOOOOXXX*X**O***X*ffffff
XXXX*fffffff***XXXXXX****X*fBfff*rrffffff
```

Board 2-3

```
➜  astar git:(master) python3 astar2.py 2-3
XXXXXXXXXXXXXXXXXXXXXXX*mmBOOOOOOO*mmmm
XXXXXXXXXXXXXXXXXXXXXXXX*mm******O*gggg
XXXXXXXXXXXXXXXXXXXXXXXXX*m****X*O*gggg
XXXXXXXXXXXXXXXXXXXXXXOOOOX*OOOOOOO*gggg
XXXXXXXXXXXXXXXXXXXXXXOXXOOOOXXXXXX*rrrr
XXXXXXXXXXXXXXXXXXXXXXOXXX*XXXXXXX*ggggg
XXXXXXOOOOOOOOOOOOOOOOXXXX**XXXX*gmmmmm
XAOOOOOXXXXXXXXXXXXXXXXXXXX*****mmmmmmm
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*wmmmmmmmmm
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*wmmmmmmmmm
```

Board 2-4

```
➜  astar git:(master) python3 astar2.py 2-4
wwww*XXXXXXXXXXXXOOOOOOOOOOOOOOOXXXXXX
wwwww*XXXXXXXXXXXOXXXXXXXXX*****OOXXXXX
wwwwww*XXXXXXXAOOOOXXXX*****wwwww*O*XXXX
wwwwww****XXXXXXXXXXX*wwwwwwwww*O*****
wwwwwwwwww**********wwwwwwwwww*O*wwww
wwwwwwwwwwwwww***wwwwwwwwwww*OO*wwww
wwwwwwwggggBg*XXX*****wwwwww*O*wwwww
wwwggggg****O**XXXXXXXX*******O*wwwww
wwggggf**XXXXOXXXXXXXXXXXXXXXOOX*wwww
wgggff*XXXXXXOOOOOOOOOOOOOOOOXXX*ggg
```

Breadth-First Search:

Board 1-1

```
➜  astar git:(master) ✗ python3 astar3_bfs.py 1-1
XXXXXXXXXXXXXXXXXX*.
XXXXXXXXXXXXXXXXXXX*
XXXXXXXXX######XXX*.
XXXXXXXXOOOAXX#OOB..
XXXXXXXXO######OXX*.
XXXXXXXXOOOOOOOOXXX*
XXXXXXXXXXXXXXXXXXX*
```

Board 1-2

```
➜  astar git:(master) ✗ python3 astar3_bfs.py 1-2
OOOOOOO#XXXXXXXXX*..
OXXXX#O#XXXXXXXXXX*.
OXXX#OO#XXXXXXXXXXX*
AXX#XO#XXOOOOOOOOOB
XXXX#OO#XOXX#XXXXXX*
XXXXX#OO#OX#XXXXXX*.
XXXXXX#OOO#XXXXXX*..
```

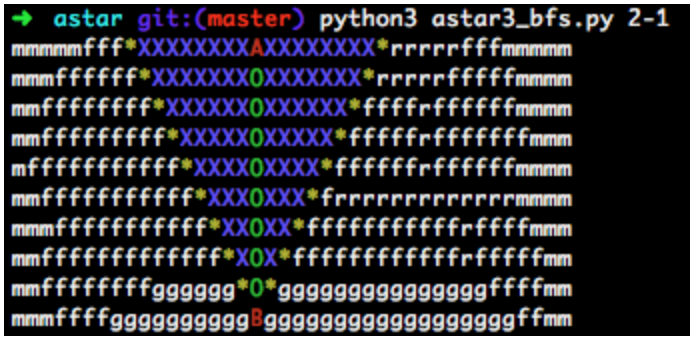
Board 1-3

```
➜  astar git:(master) ✗ python3 astar3_bfs.py 1-3
XXXXXXXXXOOOXXXXXXXX
XXXXXXXXXO#OOXXXXXXX
XXXXXXX##OO#OXXXXXXX
XXXXXX#OA#O#OXXXXXXX
XXXXXX#O#OO#OXXXXXXX
XXXXXX#OOO#XOXXXXXXX
XXXXXXX###XXOOOOOOOB
```

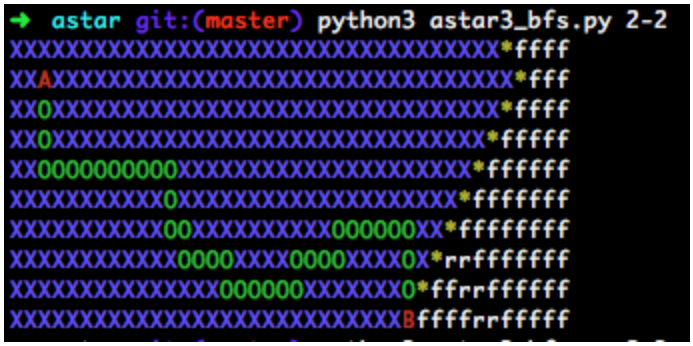
Board 1-4

```
➜  astar git:(master) ✗ python3 astar3_bfs.py 1-4
AO#X*.....#......#..
#O#X#####.#.####.#..
OO#OOOOO#.#....#....
O##O###O######.#####
OO#OB#OO#XXXX#...#..
#O####O##X##X#.#.##.
XOOOOOOXXXX#XX*#....
```

Board 2-1



Board 2-2

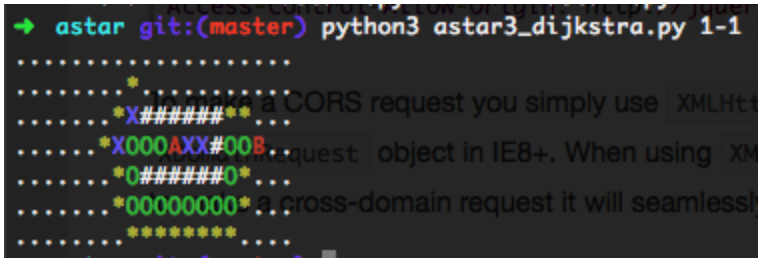


Board 2-3

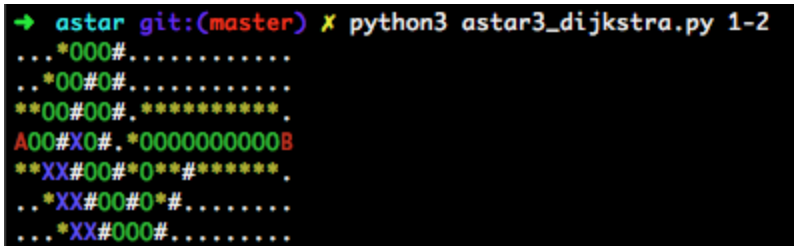
Board 2-4

Dijkstra's algorithm

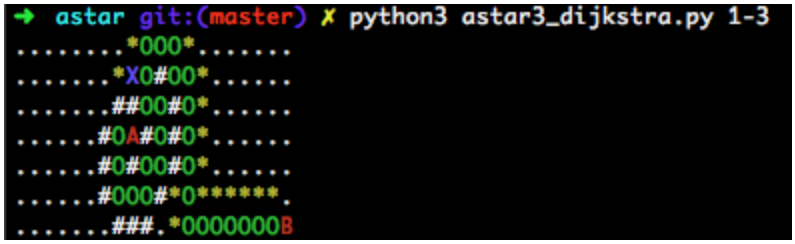
Board 1-1



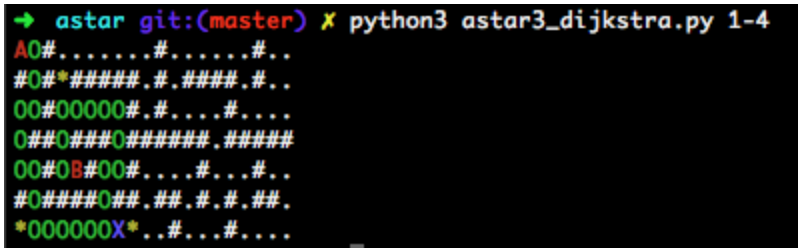
Board 1-2



Board1-3



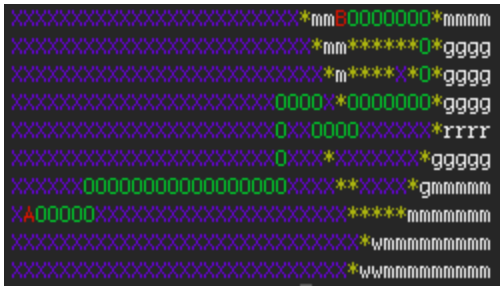
Board1-4



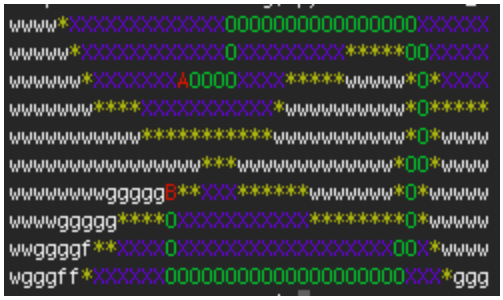
Board 2-1

Board 2-2

Board 2-3



Board 2-4



t

A3.3

Board 2.1:

With BFS, we use a FIFO queue instead of a heap, which makes it limited how far out the edges the algorithm is willing to search. It will constantly search new adjacent nodes and pushes them on the queue. The goal of the algorithm is finding B, thus it will terminate once it finds it.

Dijkstra and A* still finds the shortest path, but unlike the A*-algorithm, which uses the heuristic-function (h(N)), Dijkstra only cares about the node cost (g(N)). It updates the shortest path to a node when it finds a path that's less than the previous path to that node.

Board 2.2:

BFS finds the shortest path in this particular case. This is due to the algorithm is able to perform enough iterations that the shortest path won't be found before the least cost path is able to be found. In fact, they are found at the exact same time.

Dijkstra and A* star finds almost exactly the same result, except for a couple of nodes opened. The heuristic-function gives the advantage that one can guide the search. It will not accept a path that goes too far away from the end node.

Board 2.3:

Again, BFS is able to expand enough nodes, that it will find the least cost path in time.

Regarding Dijkstra and A*, the same conclusion is made as 2.2.

Board 2.4

Since the least cost path is a lot longer than the shortest path, BFS is not able to expand enough nodes to find the least cost path.

Dijkstra and A* find exactly the same path, and expands the same nodes. We think this is due to the fact that they want to avoid the high cost water. Because of the water nodes in the board, the heuristic function doesn't give A* an advantage.