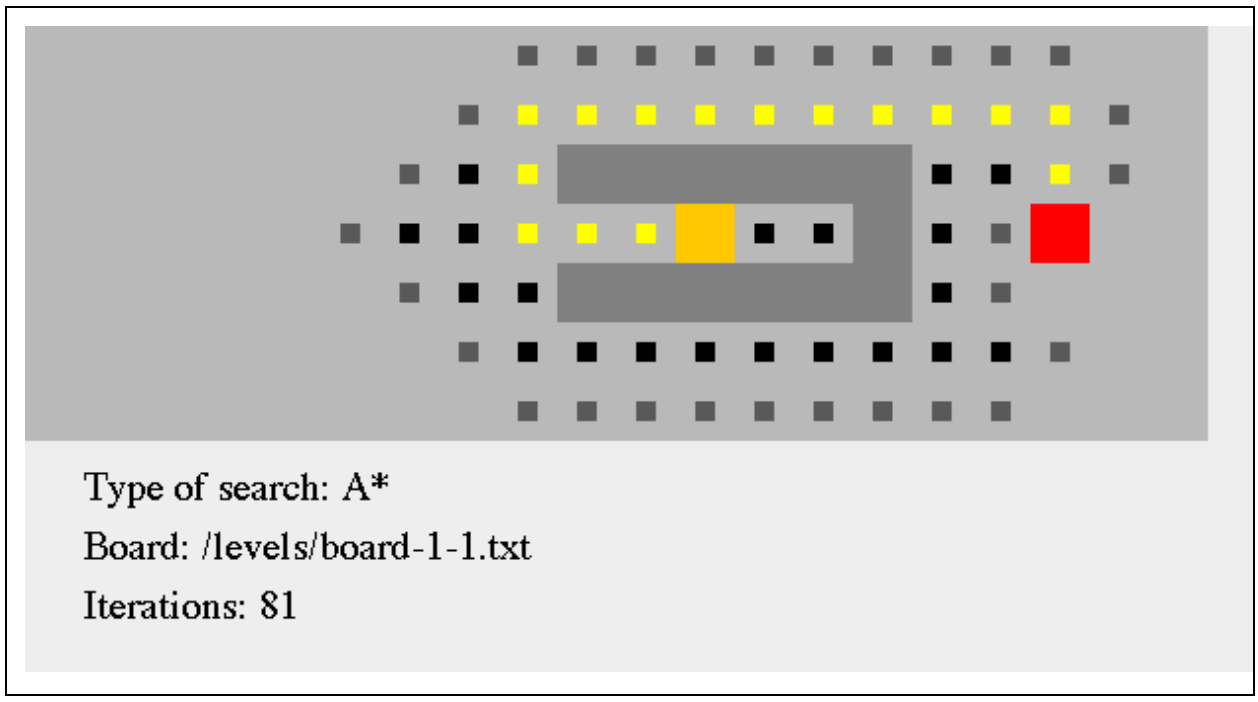
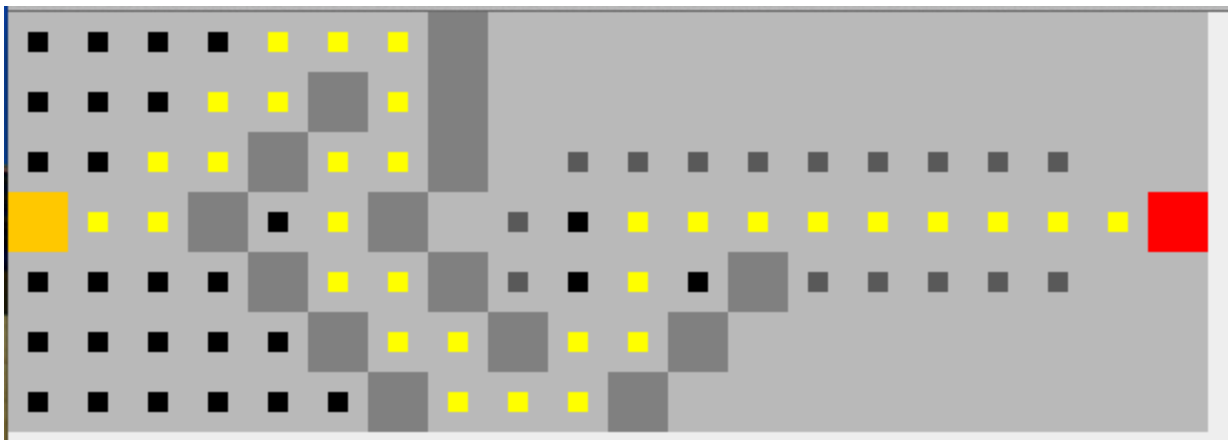
TDT4136 Introduction to Artificial Intelligence Assignment 3 – Using the A* Algorithm

Anders Lien og Simen Selseng

Problem A: Pathfinding in 2D Games

Subproblem A.1: Grids with Obstacles (1 point)

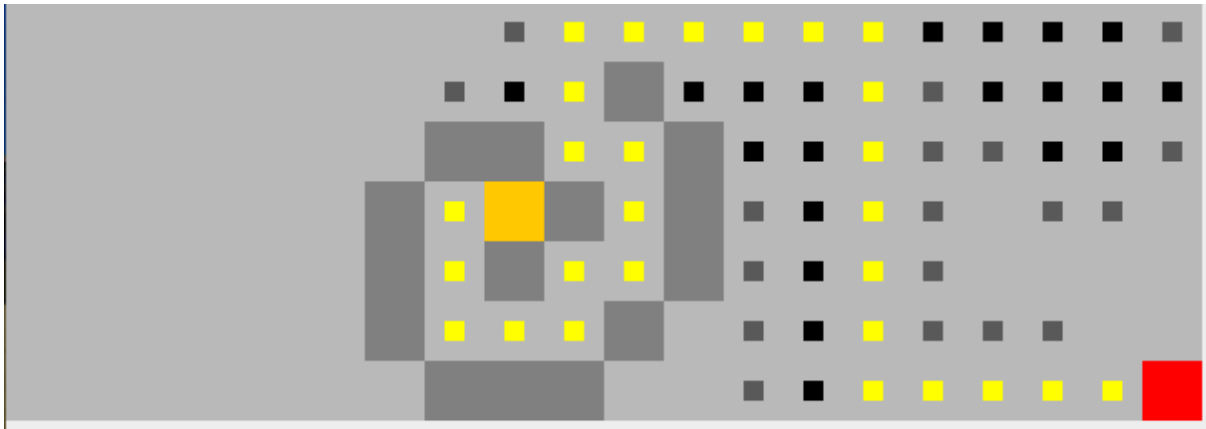




Type of search: A*

Board: /levels/board-1-2.txt

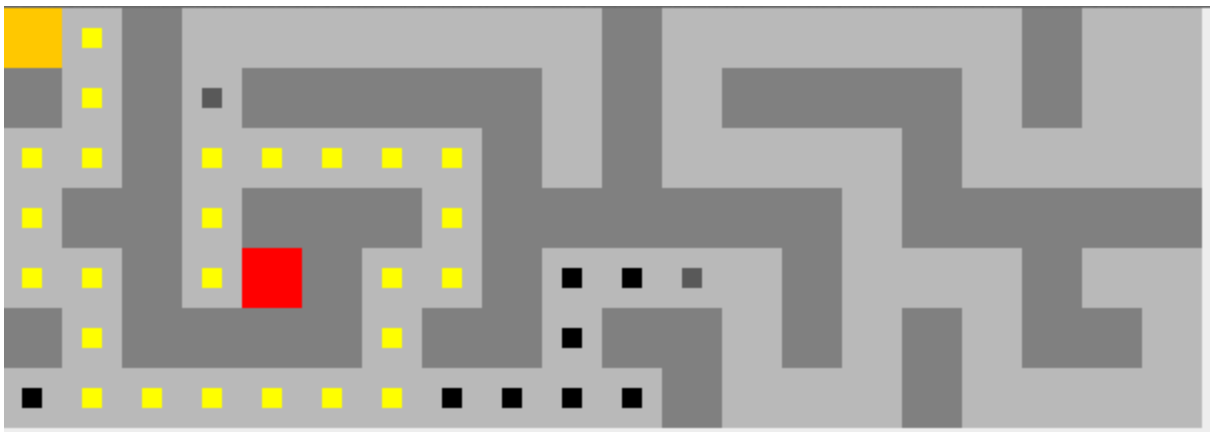
Iterations: 119



Type of search: A*

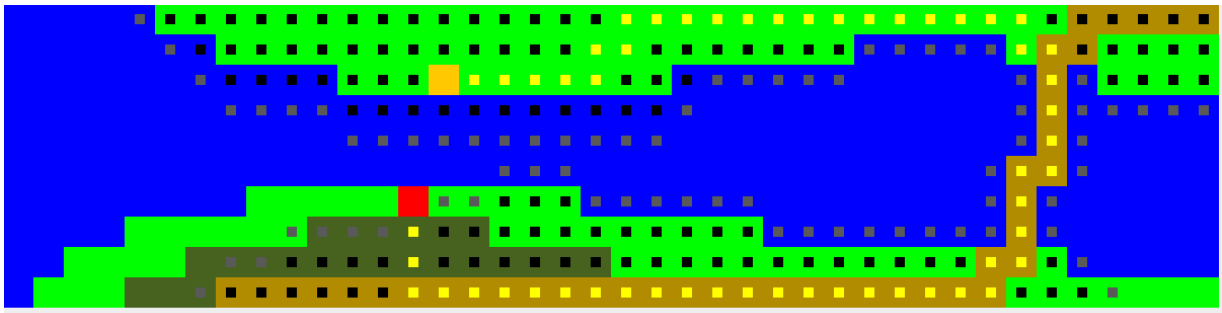
Board: /levels/board-1-3.txt

Iterations: 98

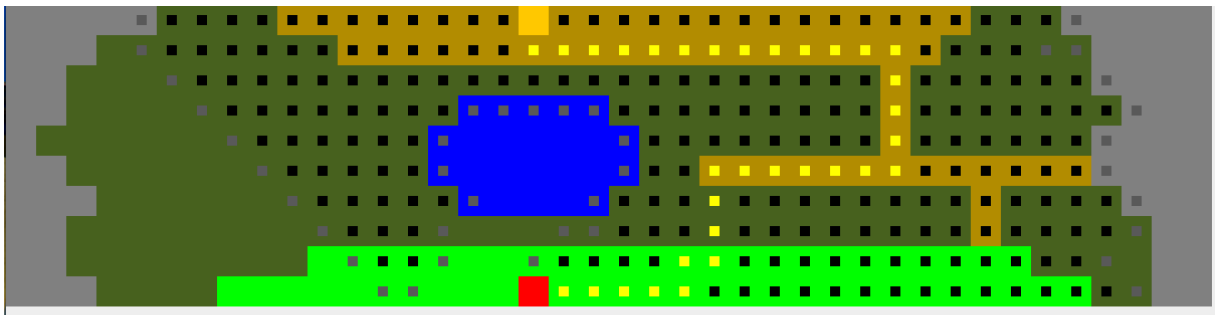


Type of search: A*
Board: /levels/board-1-4.txt
Iterations: 36

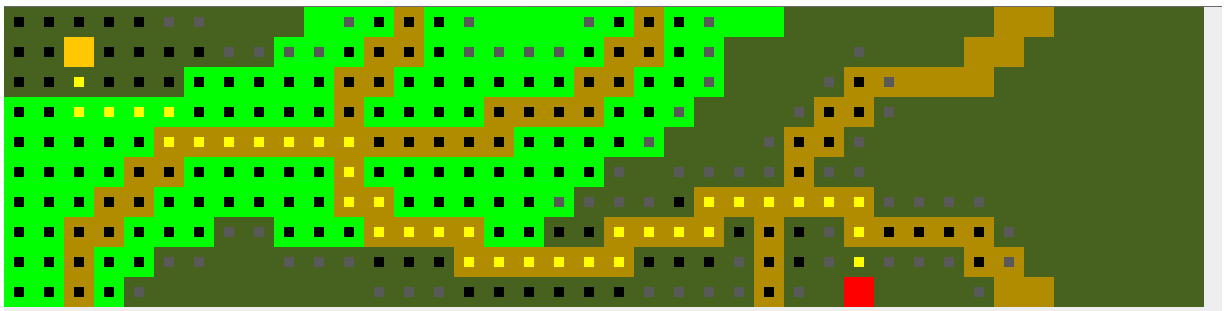
Subproblem A.2: Grids with Different Cell Costs (1 point)



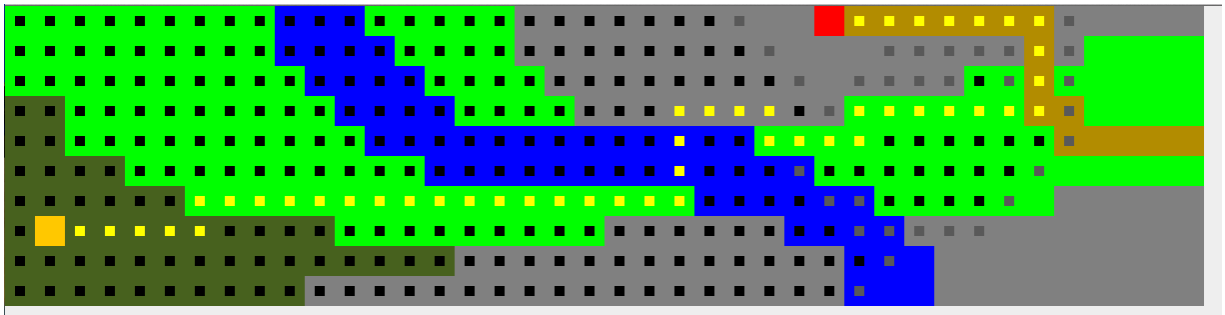
Type of search: A*
Board: /levels/board-2-4.txt
Iterations: 450



Type of search: A*
Board: /levels/board-2-1.txt
Iterations: 659



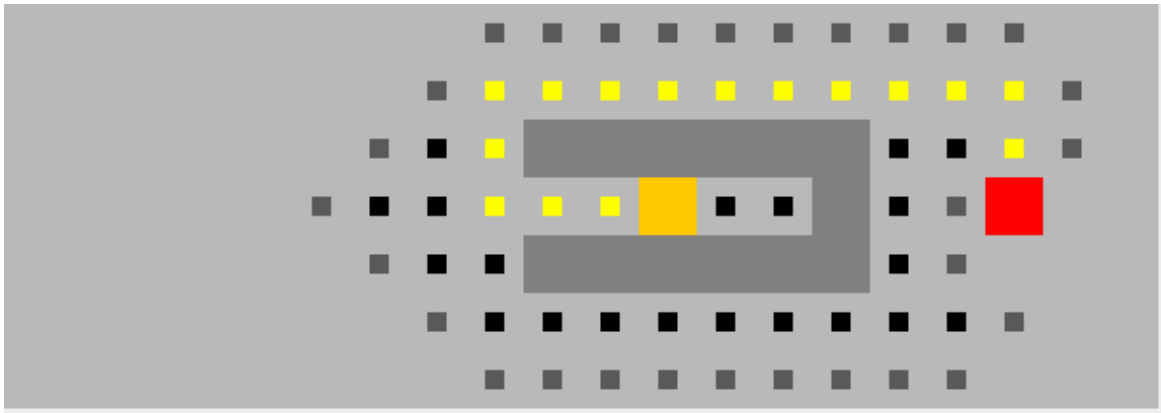
Type of search: A*
Board: /levels/board-2-2.txt
Iterations: 584



Type of search: A*
Board: /levels/board-2-3.txt
Iterations: 840

Subproblem A.3: Comparison with BFS and Dijkstra's Algorithm (1 point)

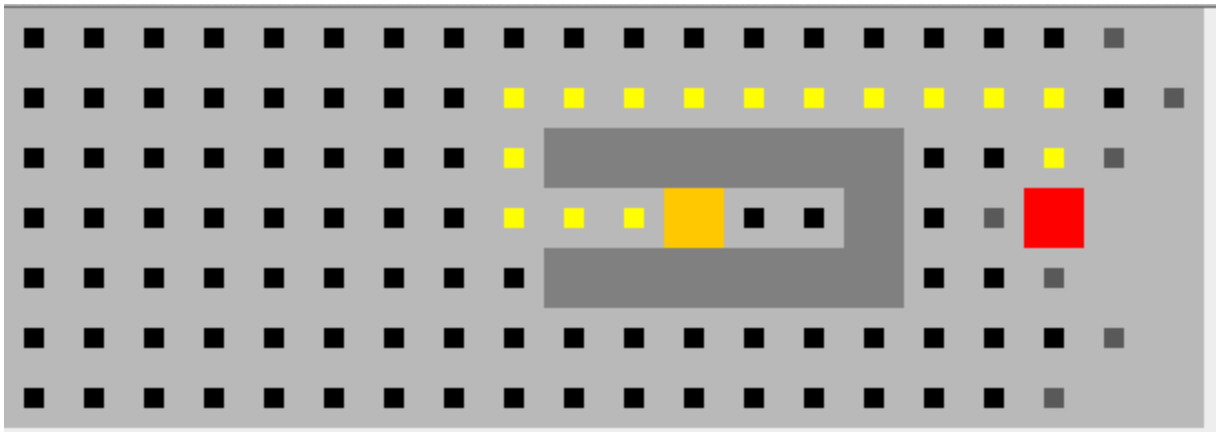
1-1



Type of search: A*

Board: /levels/board-1-1.txt

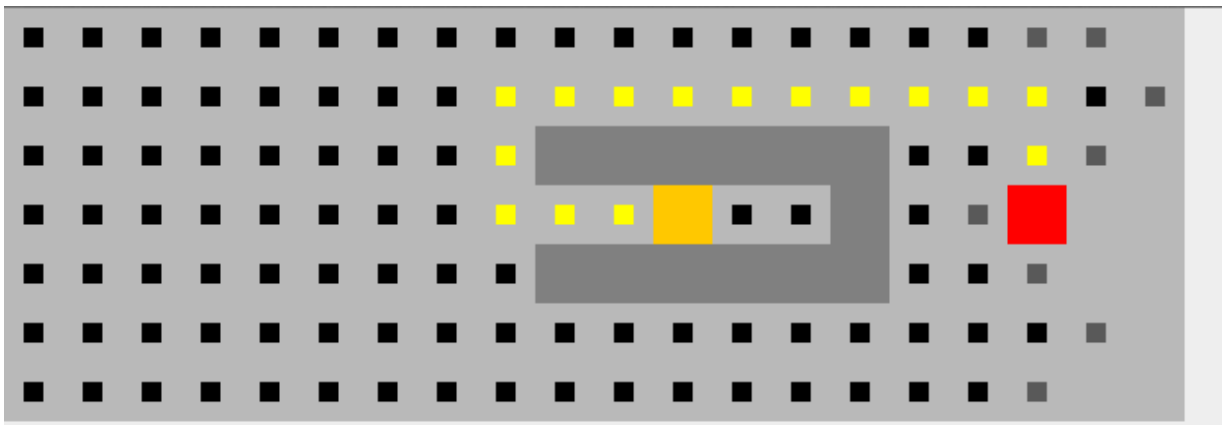
Iterations: 81



Type of search: BFS

Board: /levels/board-1-1.txt

Iterations: 261



Type of search: Dijkstra

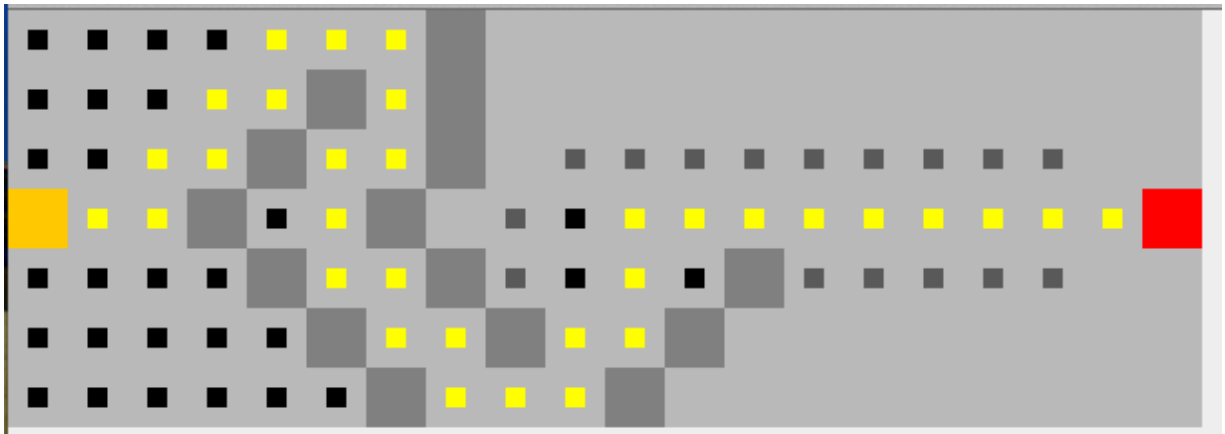
Board: /levels/board-1-1.txt

Iterations: 259

Kommentar

a) Lik korteste vei
b) A* undersøker langt færre noder enn både Dijkstra og BFS. Dijkstra of BFS undersøk nesten hele brettet før de finner målet og har derfor få noder i OPEN og mange i CLOSED. A* prosesserer mye færre noder og har ca like mange i hver.

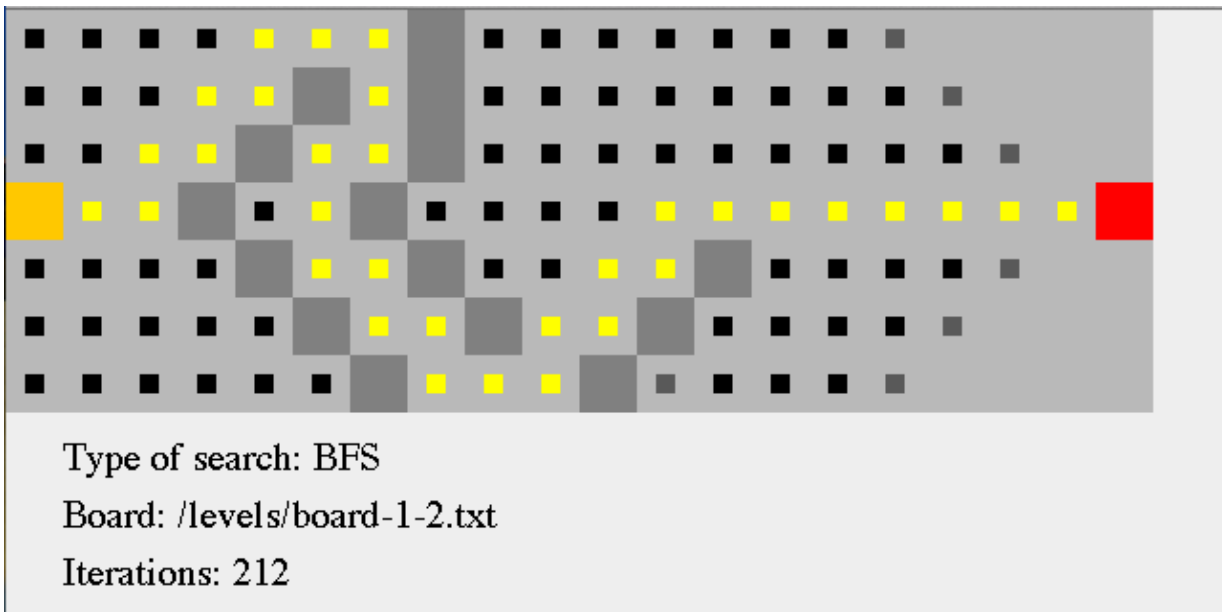
1-2



Type of search: A*

Board: /levels/board-1-2.txt

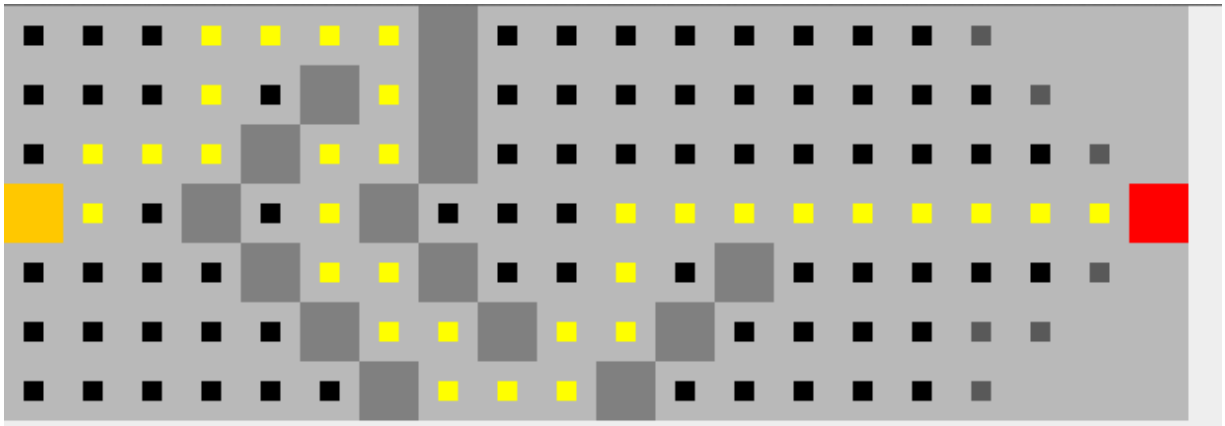
Iterations: 119



Type of search: BFS

Board: /levels/board-1-2.txt

Iterations: 212



Type of search: Dijkstra

Board: /levels/board-1-2.txt

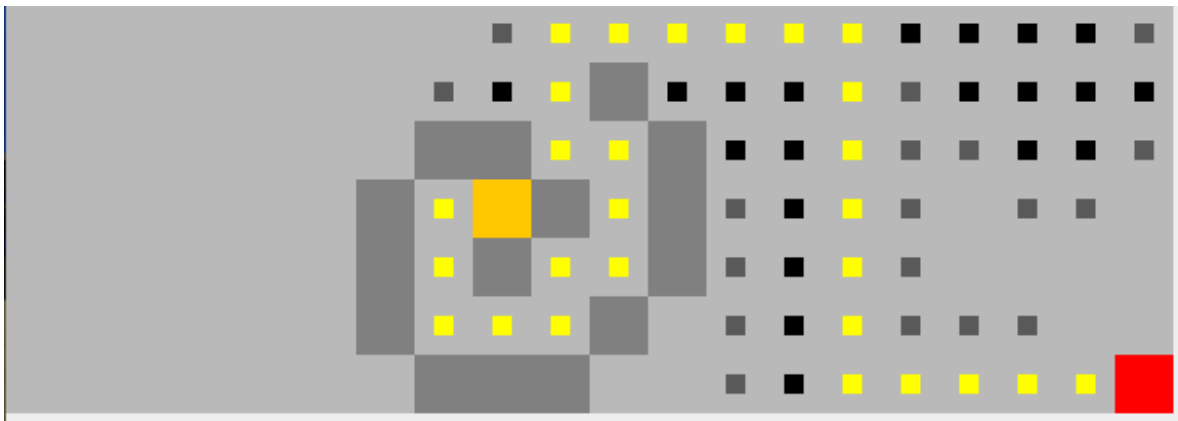
Iterations: 225

Kommentar

a) Nesten lik korteste vei (A* utforsker høyre node først siden den tar heuristikken med i regningen, Dijkstra utforsker den øverste).

b) A* utforsker mye færre noder en Dijkstra og BFS og har dermed færre noder i closed og open.

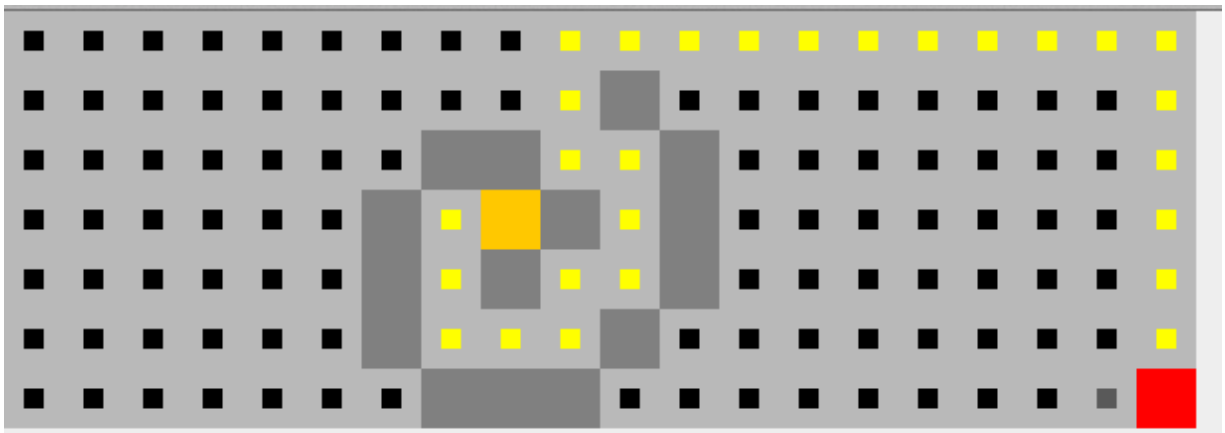
1-3



Type of search: A*

Board: /levels/board-1-3.txt

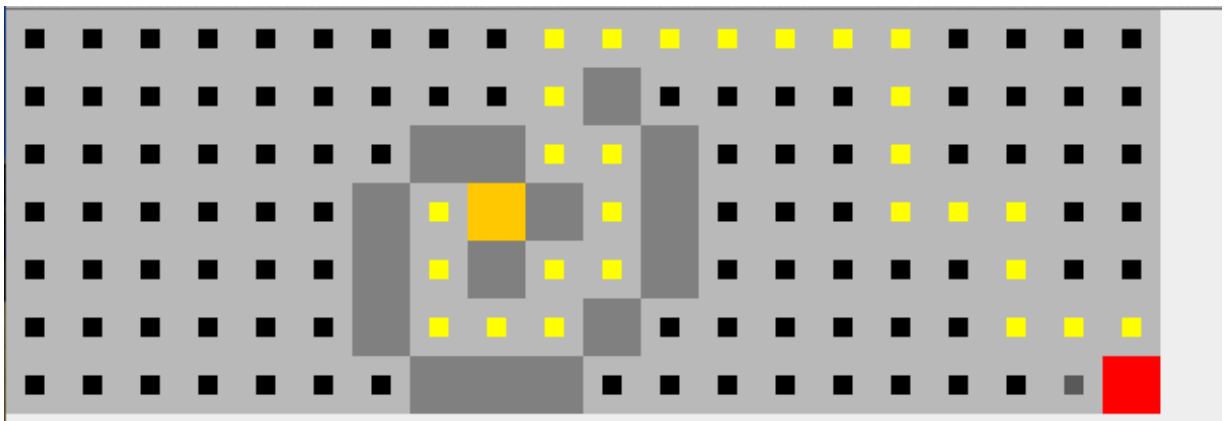
Iterations: 98



Type of search: BFS

Board: /levels/board-1-3.txt

Iterations: 279



Type of search: Dijkstra

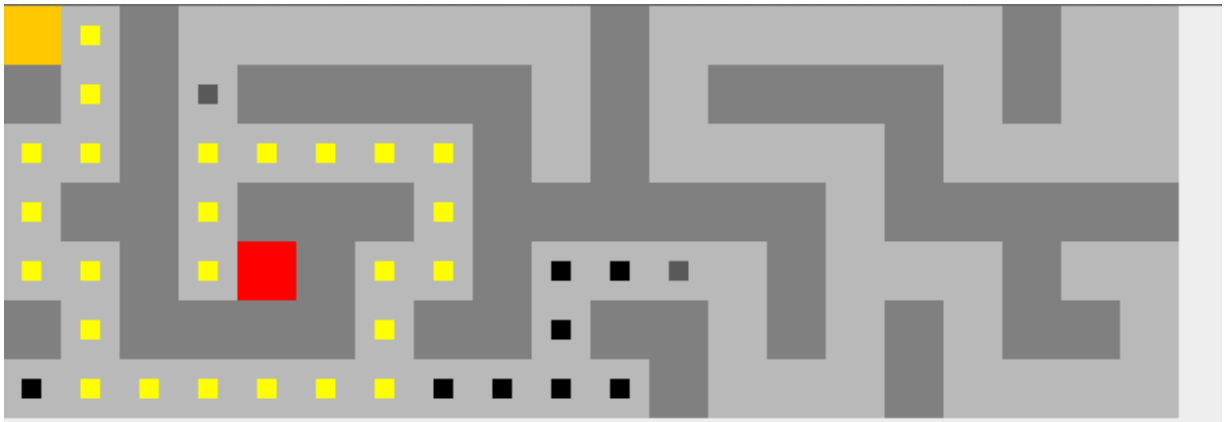
Board: /levels/board-1-3.txt

Iterations: 279

Kommentar

a) Alle algoritemen valgte forskjelleig veier til målet selv om lengden er ekvivalent.
b) Målet er nede i høyre hjørne som gjør at både BFS og Dijkstra må undersøke hele brettet før de kommer i mål. A* undersøker ca. en tredjedel av brettet. Denne forskjellen ser man også tydelig på antall iterasjoner.

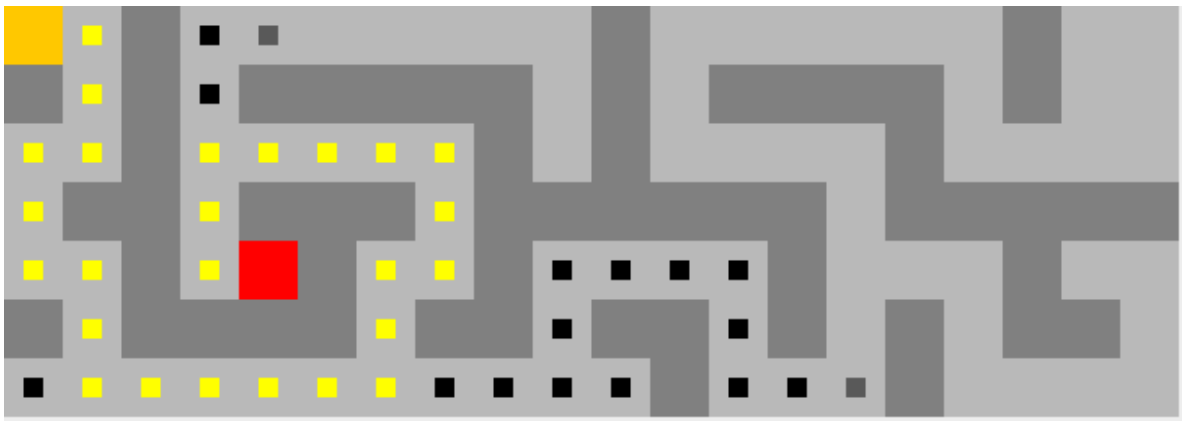
1-4



Type of search: A*

Board: /levels/board-1-4.txt

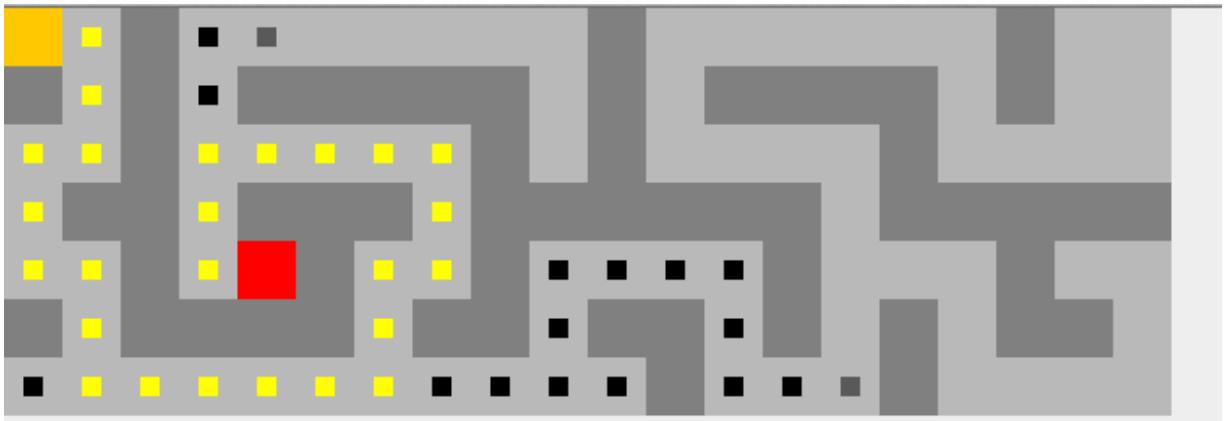
Iterations: 36



Type of search: BFS

Board: /levels/board-1-4.txt

Iterations: 43



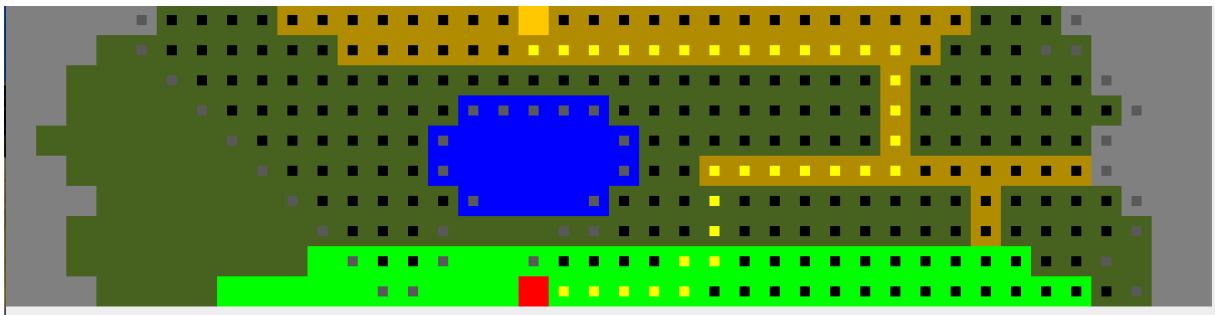
Type of search: Dijkstra

Board: /levels/board-1-4.txt

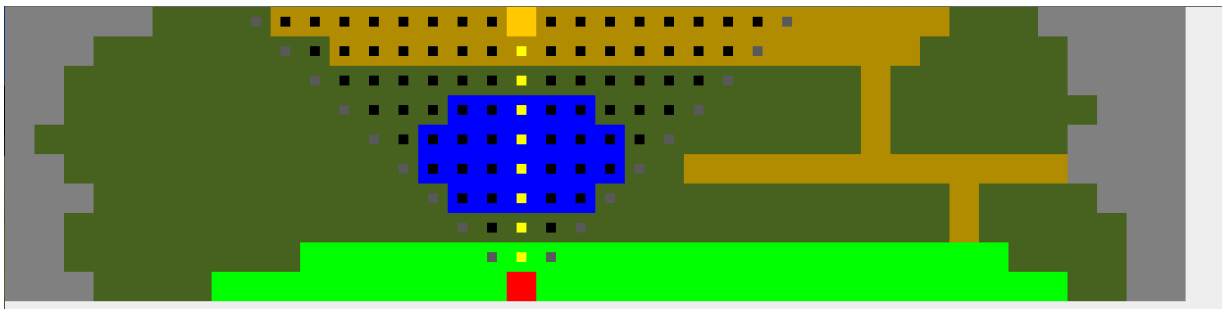
Iterations: 43

Kommentar

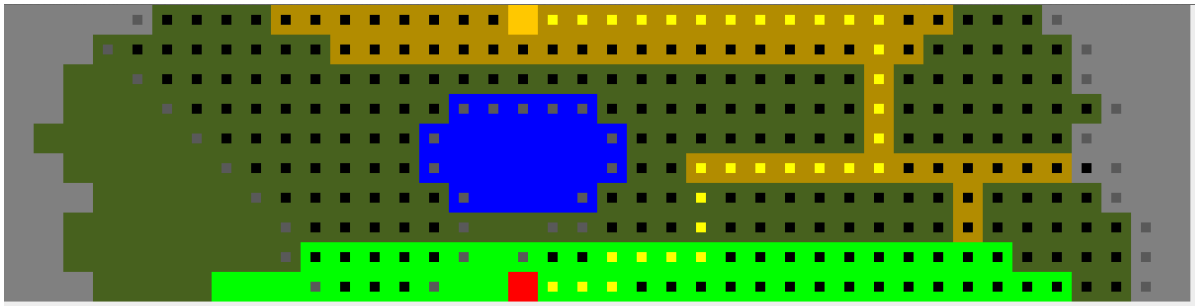
a) Alle algoritmene finner samme vei.
b) A* utforsker færre noder enn Dijkstra og BFS som begge utforsker noder i et bredde-først mønster.



Type of search: A*
Board: /levels/board-2-1.txt
Iterations: 659



Type of search: BFS
Board: /levels/board-2-1.txt
Iterations: 227

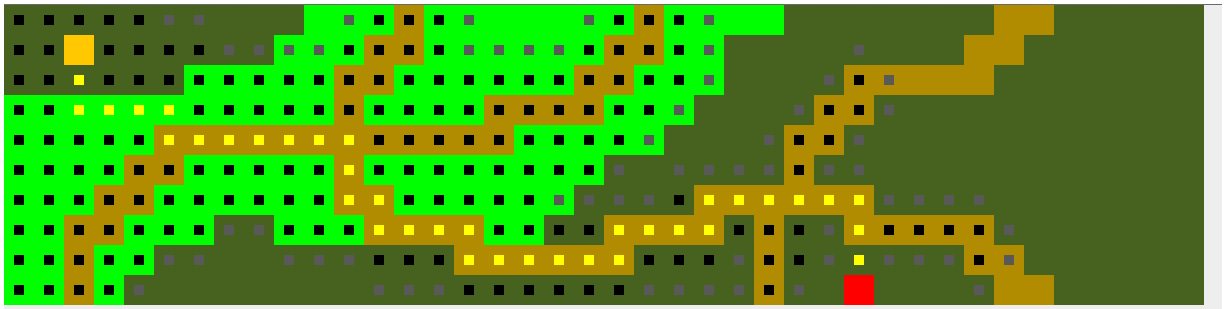


Type of search: Dijkstra
Board: /levels/board-2-1.txt
Iterations: 718

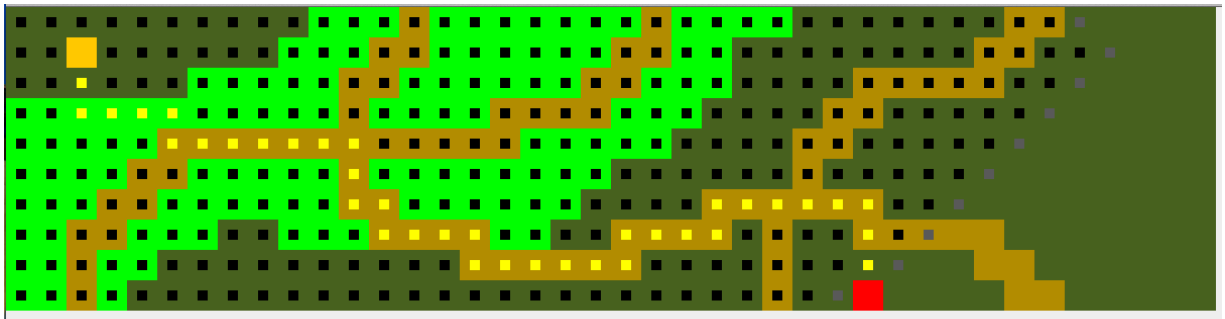
Kommentar

a) Her vil BFS feile miserabelt i å finne korteste vei siden nodene er vektet. Dijkstra og A finner to ulike veier med samme kostnad.
b) BFS vil her utforske færre noder enn de andre algoritmene siden den nesten ignorere vekten i nodene.

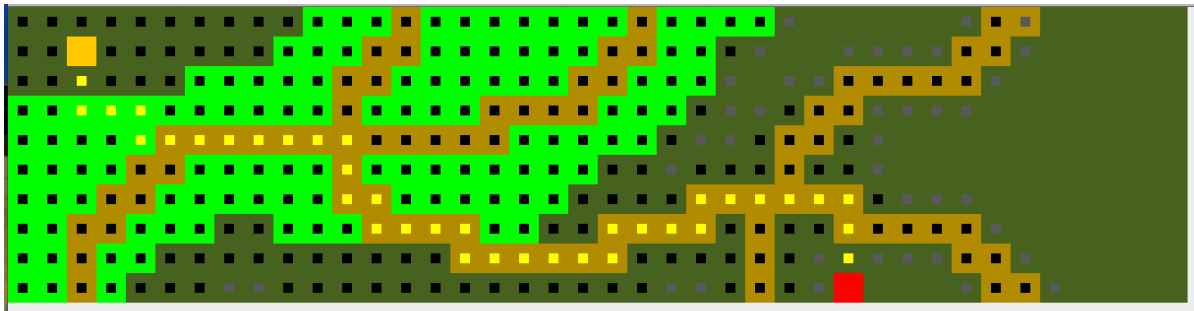
2-2



Type of search: A*
Board: /levels/board-2-2.txt
Iterations: 584



Type of search: BFS
Board: /levels/board-2-2.txt
Iterations: 895



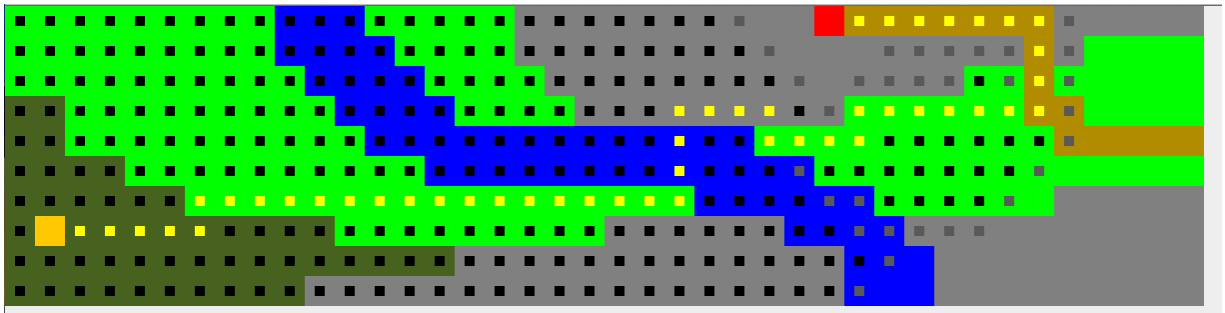
Type of search: Dijkstra
Board: /levels/board-2-2.txt
Iterations: 788

Kommentar

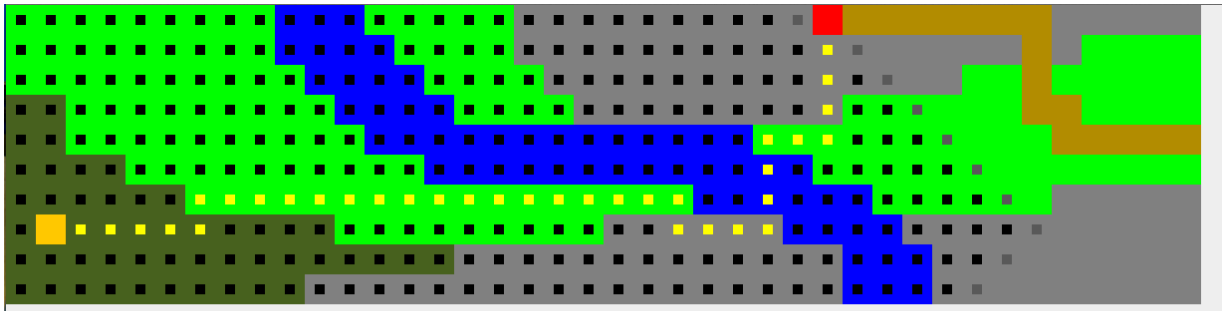
a) Alle algoritmene finner samme ekvivalente vei. Algoritmene følger den samme veien t tross for ulikhetene, fordi de alle bruker g til å bestemme hvilken node som skal være parent.

b) BFS utforsker omtrent hele brettet før den oppdager sluttnoden, den har altså veldig mange noder i closed. A* og Dijkstra gir ganske like resultater, sistnevnte har litt flere besøkte noder.

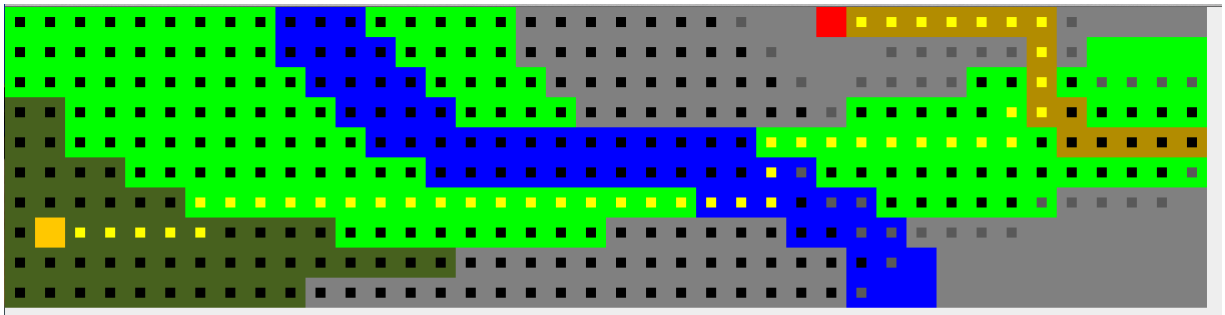
2-3



Type of search: A*
Board: /levels/board-2-3.txt
Iterations: 840



Type of search: BFS
Board: /levels/board-2-3.txt
Iterations: 857



Type of search: Dijkstra
Board: /levels/board-2-3.txt
Iterations: 892

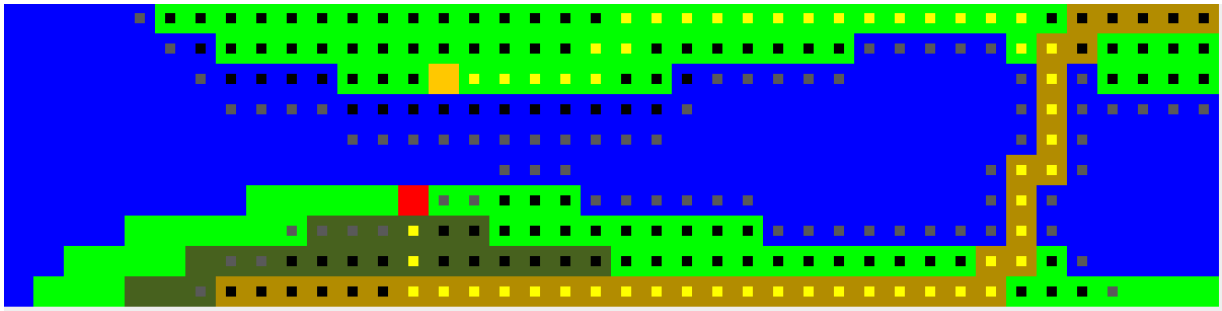
Kommentar

a) Algoritmene begynner likt, men velger alle forskjellige, men ekvivalente, veier over elv
BFS finner ikke fjellveien og velger en dårlige ruter over fjellet.

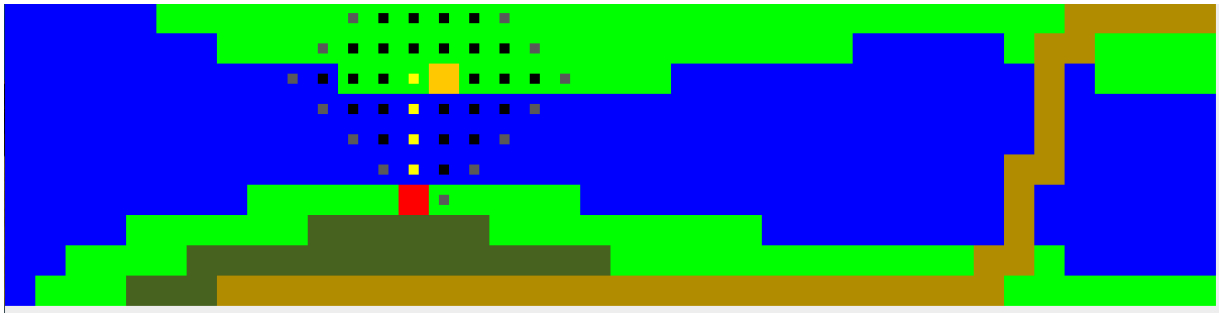
b) Igjen finner BFS målet tidlig og stopper søket før den finner fjellveien. Dermed har BF
færrest noder i både Closed og Open. A* undersøker litt flere noder, mens Dijkstra

undersøker nesten hele brettet før den kommer i mål.

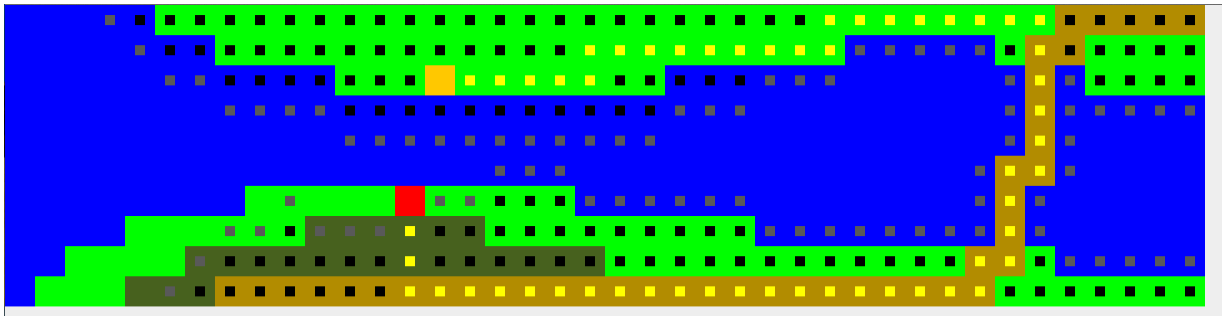
2-4



Type of search: A*
Board: /levels/board-2-4.txt
Iterations: 450



Type of search: BFS
Board: /levels/board-2-4.txt
Iterations: 87



Type of search: Dijkstra
Board: /levels/board-2-4.txt
Iterations: 479

Kommentar

a) A* og Dijkstra har valgt den samme ruten over broen. BFS derimot, finner målet tidlig og velger den korteste veien utifra de nodene den har besøkt allerede, uten å se på

terrenget. Som vi ser, gjør dette at BFS velger den korteste veien i luftlinje.

b) BFS finner målet tidlig og utforsker derfor få noder. A* og Dijkstra utforsker store dele kartet, men holder seg borte fra vannet.