

Elif Çalışkan 2016400183
CMPE480 Homework1 Report

Bfs

In breadth first search, the positions are held in hashset and queue is used for traversing the graph. parentSet is used for finding expanded nodes (if there is a node, its path until that node was expanded). First, start node is added to queue, then for each node that was polled, its children are added to queue if they weren't traversed.

Level1 output:

11 35 7 7
RRDRRRD

Level2 output:

19 19 11 11
DLDLDRRULDR

Level3 output:

18 26 10 10
LDLULURRDD

Level4 output:

33 119 19 19
ULDRURRRRUURRDDDRU

Dfs

In depth first search, the positions are held in hashset and stack is used for traversing the graph. First, start node is added to stack, then for each node that was popped, its children are pushed to stack if they weren't traversed.

Level1 output:

105 72 63 63
RRDLLULURDLLURDRRDLLURRDRURDRRRDLULULLDRRRULDLURRDLULLLDRRRDL

Level2 output:

28 32 19 16
DLDLDRULURDRDLUR

Level3 output:

26 19 16 14

LLDLRRUULDRDD

Level4 output:

102 60 60 60
ULDRULDRURULDLURRDLULDRRRRUURRRDRULLLLDRURRRDRDLLUURDRDLDRU

Ucs

In uniform-cost search, the positions are held in hashset and priority queue is used for traversing the graph. parentSet is used for finding expanded nodes (if there is a node, its path until that node was expanded). First, start node is added to queue, then for each node that was polled, its children are added to queue if they weren't traversed. Every node that wasn't traversed, still stays in the queue and search continues with the least expensive node. CostComparator is used in order to create a min heap.

Level1 output:

10 56 8 8
DRRRRRRD

Level2 output:

19 25 11 11
D DLLDRRULDR

Level3 output:

18 33 11 10
LDLULURRDD

Level4 output:

33 169 21 19
ULDRURRRRUURRRDDDRU

Greedy

In greedy search priority queue is used for traversing the graph(it has at most 4 nodes since it is greedy and we try to find the best local solution). First, start node is added to priority queue, then for each node that was polled, the queue is cleared and its children are added to queue since the aim is finding the best local solution. DistComparator is used in greedy algorithm. The heuristic is: find the node with least sum of distance to goal from that node and cost of the step. But since it goes into loop, it is not a good heuristic.

Level1 output:

11 7 7 7
RDDRRDR

Level2 output:

Loop

Level3 output:

Loop

Level4 output:

Loop

A*

In a* search priority queue is used for traversing the graph(it has at most 4 nodes since it is greedy and we try to find the best local solution). First, start node is added to priority queue, then for each node that was polled, the queue is cleared and its children are added to queue since the aim is finding the best local solution. AsComparator is used in a* algorithm. The heuristic is: find the node with least sum of distance to goal from that node and cost of arriving to that step from the beginning. But since it goes into loop, it is not a good heuristic.

Level1 output:

11 7 7 7
RDDRDR

Level2 output:

Loop

Level3 output:

Loop

Level4 output:

Loop

How to run:

proj1 <path of level.txt> dfs

But I used eclipse run configuration for it to be easier to see since command prompt is not a console and written values don't show.