

CMPE 480

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

PROJECT #1

Due 04.11.2019

In this project, you will implement depth-first, breath-first, uniform-cost, greedy and A* search algorithms for a grid-based puzzle.

The Puzzle Definition

In this puzzle, there is an agent on a grid-world. The agent can occupy a single cell or two adjacent cells. The state of the agent can be represented by its current position and orientation (single cell, horizontal or vertical). It starts at a single cell. The aim is to move the agent to a designated goal position.

Four actions can be performed: up, down, right, left. If the agent is at a single cell, the actions cause the agent to move in the intended direction and occupies two cells. The orientation of the agent depends on the action taken.

If the orientation of the agent is horizontal, right/left actions move the agent in the intended direction and changes the orientation to single. Up/down actions move the agent one cell in the intended direction.

If the orientation of the agent is vertical, up/down actions move the agent in the intended direction and changes the orientation to single. Left/right actions move the agent one cell in the intended direction.

The **cost** of the action is **3 (three)** for taking left or right action when the orientation is horizontal and up or down action when the orientation is vertical. For all other cases, the cost is **1 (one)**.

All possible state transitions are listed at the below table.

		Actions			
Orientation	Current state	Down	Up	Left	Right
single					
horizontal				 cost=3	 cost=3
vertical		 cost=3	 cost=3		

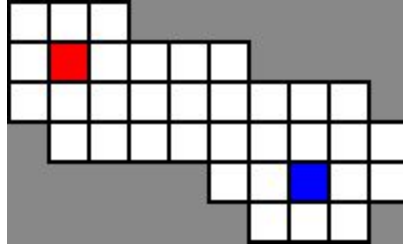
The environment is given as a text file. The first line includes the width and height (max 20x20) of the grid separated by a space. The following lines include the state of each grid cell row by row. Each cell is represented by a single character: 'o' (available cell), ' ' (space, empty cell), 's' (start position), 'g' goal position. Example file is given below.

```

10 6
ooo
osoooo
ooooooooo
ooooooooo
  ooooo
    oogoo
      ooo

```

This input refers to the following grid where *white* represents the allowed cells, *red* is the agent and the *blue* is the goal position. The agent is not allowed to move onto the empty cells represented by *gray*.



The agent starts at a single cell. Your program will search for the sequence of actions, which moves the agent to the goal cell and having the orientation “single”. The agent is only allowed to be on available cells, start position or goal. It is not allowed to be on empty cells (represented by the spaces in the input file).

See <http://komputer.com.tr/demo/ersin/cmpe480/> for an interactive demo of the puzzle.

See <http://komputer.com.tr/demo/ersin/cmpe480/index.js> for JS implementation of puzzle logic.

Download the levels from <http://komputer.com.tr/demo/ersin/cmpe480/levels/levels.zip> and use these levels in your report.

How to run

Your code will be called with two parameters: the input environment (level) file and the algorithms to use (dfs, bfs, ucs, gs, a*). For example

```
proj1 levels/level1.txt dfs
```

should perform depth-first search on the input environment defined in “levels/level1.txt” file where proj1 is the name of the executable. The algorithms and their corresponding parameter is listed in the following table:

Algorithm	Command line parameter
Depth-first search	dfs
Breadth-first search	bfs
Uniform-cost search	ucs
Greedy search	gs
A* search	a*

Your program should output the following values:

1. Path cost of the solution
2. The number of expanded nodes (the explored nodes)
3. The depth of the generated tree
4. The depth of the solution node
5. The solution: sequence of actions to reach the goal state (use the initial letters **L**eft, **U**p, **R**ight, **D**own, no space between them)

You should use the standard output, not a text file. A sample output is

```
11 35 7 7
RRDRRRD
```

Where 11 is the cost of the solution, 35 is the number of expanded nodes, 7 is the depth of the tree, 7 is the depth of the solution and `RRDRRRD` is the solution. The depth of the tree is zero when there is only the root cell.

Important Note: When expanding a node in the tree, use the action order **left**, **up**, **right**, **down**. So left action is evaluated first and down action is evaluated last. This is important because this order changes both the generated tree and the solution.

Programming Language

You can use any programming language of your choice. However, since your code may also be checked if it is compiled into the executable in your submission, you should include the details of the compilation process in your report, if you use a not-so-common programming language or setup.

Scripting languages like javascript are also welcome. If you use JS, feel free to use the codes in the interactive demo for state representation and transition. However your program should run on NodeJS, not the browser.

What to Submit?

You will submit both **source code**, **compiled executable** (if available, name it `proj1.exe` for windows or `proj1` for linux) and a **short report** in PDF format (explaining your solution and heuristics). In the report, you should also include the output values (list in How to Run section) and discuss on these results (what are the reasons for that results). For the heuristic, you should also justify your decision.

Submit a single zip file with your student id as the filename. Don't forget to include your name and student ID in the report also. Don't include the input puzzle level files.

How to Submit?

Send a zip file of including source code, executable and report to ersin@komputer.com.tr with the subject "**CMPE 480 PROJECT#1 2xxxxxxxx**" (where 2xxxxxxxx is your student number).

Late submissions will be penalized. Each day after the due date causes %30 drop in your grade. You should submit **before 04.11.2019 23:59** to receive full credit.

Cheat Policy

You are not allowed to cheat from either from your classmates or the internet. Read the department policy about what is considered as cheating. You will receive F and a disciplinary action if you are suspected to cheat.

Questions

Use cmpe480@listeci.cmpe.boun.edu.tr for your questions on the project.