

Assignment 1

Uninformed & Informed Search

Max possible score:

- 4308: 100 Points [+40 Points EC]
- 5360: 100 Points

Task 1

Max: [4308: 100 Points, 5360: 100 Points]

Your task is to build an agent to solve a modified version of the 8 puzzle problem (called the Expense 8 puzzle problem). The task is still to take a 3X3 grid on which 8 tiles have been placed, where you can only move one tile at a time to an adjacent location (as long as it is blank) and figure out the order in which to move the tiles to get it to a desired configuration. However now the number on the tile now also represents the cost of moving that tile (moving the tile marked 6 costs 6).

Your program should be called `expense_8_puzzle` and the command line invocation should follow the following format:

`expense_8_puzzle.py <start-file> <goal-file> <method> <dump-flag>`

- `<start-file>` and `<goal-file>` are required.
- `<method>` can be
 - bfs - Breadth First Search
 - ucs - Uniform Cost Search
 - dfs - Depth First Search [Note: This part is EC for CSE 4308 students]
 - dls - Depth Limited Search (Note: Depth Limit will be obtained as a Console Input) [Note: This part is EC for CSE 4308 students]
 - ids - Iterative Deepening Search [Note: This part is EC for CSE 4308 students]
 - greedy - Greedy Search
 - a* - A* Search (Note: if no `<method>` is given, this should be the default option)
- If `<dump-flag>` is given as true, search trace is dumped for analysis in `trace-<date>-<time>.txt` (Note: if `<dump-flag>` is not given, assume it is false)
 - search trace contains: fringe and closed set contents per loop of search (and per iteration for IDS), counts of nodes expanded and nodes

Both start file and goal file need to follow the format as shown here:

- [Sample Start file](#)
- [Sample Goal file](#)

Your output needs to follow the format given in the example here:

For:

`expense_8_puzzle.py start.txt goal.txt a* true`

The output should appear as follows:

Nodes Popped: 97
Nodes Expanded: 64
Nodes Generated: 173
Max Fringe Size: 77
Solution Found at depth 12 with cost of 63.

Steps:

Move 7 Left
Move 5 Up
Move 8 Right
Move 7 Down
Move 5 Left
Move 6 Down
Move 3 Right
Move 2 Right
Move 1 Up
Move 4 Up
Move 7 Left
Move 8 Left

In addition, the search trace should have all the information contained in the file given [here](#). (The format is not important, the information is)

Note: for both greedy and A* search you need to come up with a acceptable heuristic (Hint: Consider a modified version of h2 as discussed in class)

Note: Implementing DFS, DLS and IDS is required for CSE 5360 students but optional for CSE 4308 students (and carries upto 40 points EC)

How to submit

Implementations in C, C++, Java, and Python will be accepted. Points will be taken off for failure to comply with this requirement unless previously cleared with the Instructor.

Create a ZIPPED directory called <net-id>_assmt1.zip (no other forms of compression accepted, contact the instructor or TA if you do not know how to produce .zip files).

The directory should contain the source code for the task (no need for any compiled binaries). Each folder should also contain a file called readme.txt, which should specify precisely:

- Name and UTA ID of the student.
- What programming language is used for this task. (Make sure to also give version and subversion numbers)
- How the code is structured.
- How to run the code, including very specific compilation instructions, if compilation is needed. Instructions such as "compile using g++" are NOT considered specific if the TA needs to do additional steps to get your code to run.
 - If your code will run on the ACS Omega (not required) make a note of it in the readme file.
- Insufficient or unclear instructions will be penalized.
- Code that the TA cannot run gets AT MOST 75% credit (depending on if the student is able to get it to run during a Demo session).

