COEN 6321 - Final Project (default)

Introduction

A few years ago, a puzzle was released and the creator offered a £2,000,000 prize to whoever was able to solve it by a given deadline. No one could find the solution, and no perfect solution has been found to this date. Your project is to solve such a puzzle. The original puzzle is called “Eternity II” because it’s supposed to take an eternity to solve it via brute-force methods.

Overview

The puzzle here is an edge-matching puzzle. The pieces are triangular. The objective is to match as many pieces as possible to one another. The black edges of pieces must be at the borders/extremities of the puzzle board. That’s it. You can use any technique you’ve seen in class from evolutionary algorithms, heuristic learning algorithms or a combination. You can work individually or in a team of 2 people.

Deliverables

We will give you 3 puzzles: (i) a trivially easy one to test your algorithm, (ii) a challenging one [unique to every team] and (iii) a very difficult one. You will need to attempt to solve all 3.

You may write your program in any software. You can use EA and/or HLA packages available online. You may NOT use packages created to solve this specific problem – edge-matching puzzle problem. You will be asked to demonstrate your code in person; exception: students submitting a Jupyter notebook don’t need to since I can run the code myself if it’s well documented. You will need to provide the result(s) you obtain for every puzzle.

Marks will be proportional to how fit (# of edge matching) your solution is given how many times you’ve evaluated the problem. For puzzle “iii”, this will be relaxed for thoughtfully creative methodologies – even if they don’t find a very good solution. Fitness will equate to number of edges NOT matching for an optimal fitness value of 0 (or the inverse if you think it necessary).

Undergraduate students will write a program to solve the puzzle and a short report. The report will show the problem, explain the methodology (in writing and with a pseudo-code) and then show the results. A discussion and conclusion to wrap it up. The report should be around 3 pages; it can be part of the Jupyter notebook should you chose to do it that way.

Graduate students will write a program to solve the puzzle and a term paper. The paper will begin with a review of the problem and show a few algorithms while naming their advantages/disadvantages. Then, show the problem, explain the methodology in detail (writing and pseudo-code + expectations based on previous

work) and the show the results with appropriate analysis. This is followed by a thoughtful discussion, conclusion and future work to wrap it up. The paper should be at least 6 pages. in IEEE conference paper format, preferably in LaTeX.

Puzzle i

Let's give an example of what the puzzle looks like and then what you will be receiving for puzzles "ii" and "iii". Here are the pieces. They're triangles with 3 colours. Each piece can be rotated but not flipped/mirrored:

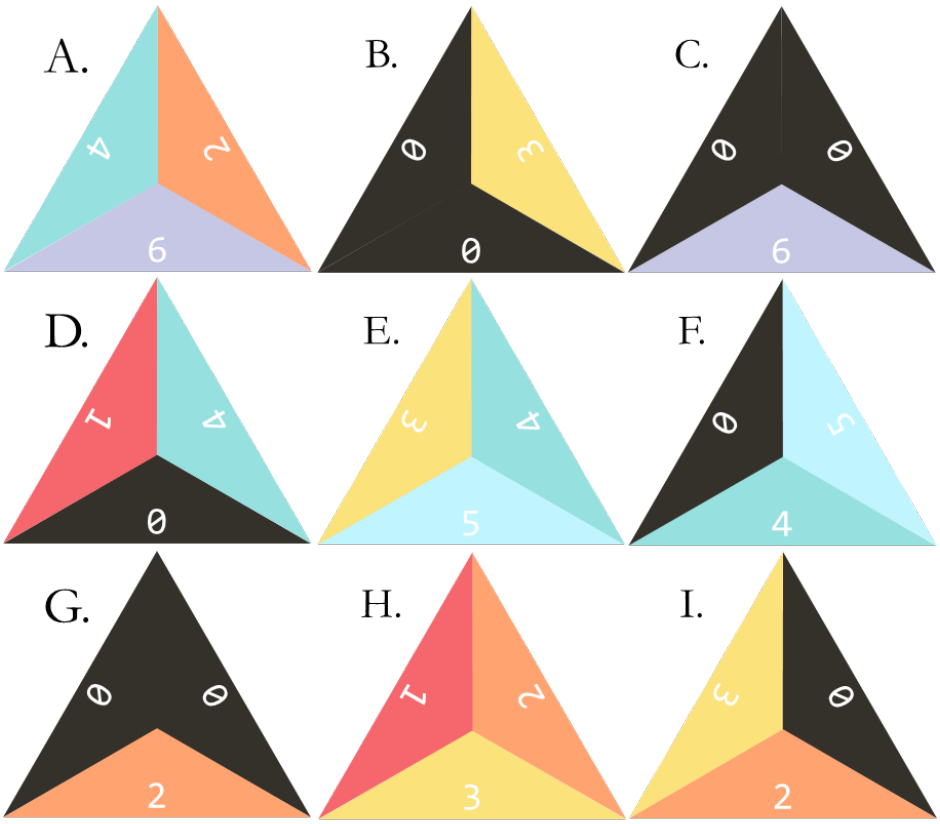

Figure 1: Puzzle pieces.

The colours are simply to help with the visual cue. Each colour corresponds to a number. Your program should not have a limit as to how many colours/numbers we have in a puzzle. Here are the colours/numbers used in puzzle "i":

Figure 2: Colours! Notice that we could've continued...

You have to take the pieces in Figure 1 and arrange them so that the puzzle piece edges match and that the black "0"s are around the edges of the final board. Pretty straightforward! The solution is on the next page. Don't look just yet!

Puzzle i solution

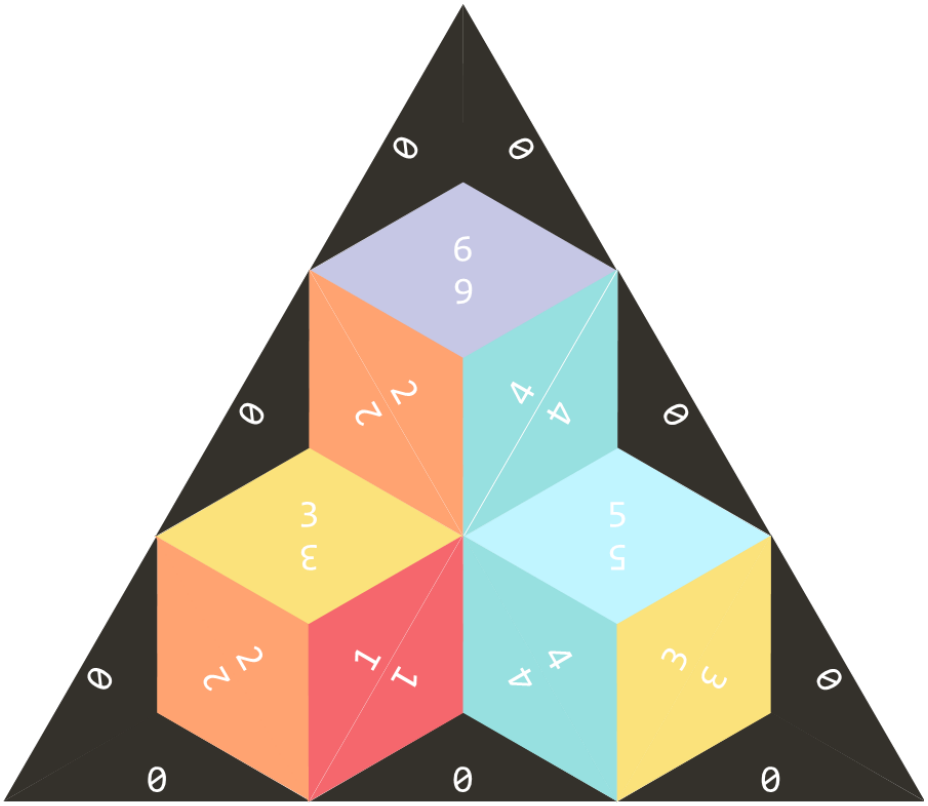

Figure 3: Puzzle “i” solution. Notice the black at the edges.

We will be sending or posting the pieces in this format: Piece letter, edge number(s) in clockwise order. Some pieces may be duplicates. Please verify with what you see in Figure 1.

A, 4, 2, 6
B, 0, 3, 0
C, 0, 0, 6
D, 1, 4, 0
E, 3, 4, 5
F, 0, 5, 4
G, 0, 0, 2
H, 1, 2, 3
I, 3, 0, 2

A solution would show which piece (letter) goes where and in what orientation.

That’s all, good luck!

- EOF -