

Department of Electrical & Computer Engineering (ECE), Concordia University

APPLIED MACHINE LEARNING & EVOLUTIONARY ALGORITHMS

COEN 432 (6321): Fall 24

Assignment 1: Edge-Matching Puzzle (due date: Fri 11 Oct @ 23hr55, via Moodle)

Problem Description. Your program must attempt to solve an 8x8 square puzzle containing 64 square pieces. The puzzle pieces are provided in an input file. The input file is 64 pieces in a random arrangement, with eight 4-digit numbers per line (for a total of 8 lines). Note that when you create the first population of puzzles (i.e., candidate solutions), each puzzle must contain all the 64 pieces but, at least in the first population, the positions and orientations of the 64 pieces should be different, from one puzzle to another.

Puzzle Representation. There are 64 pieces in the puzzle, arranged in an 8x8 square. Each tile has 4 edges and is represented by 4 numbers. Each number represents a motif (think of it as a pattern or a colour) and there are a total of 7 motifs. The first number represents the top edge, the second number represents the right edge, the third number represents the bottom edge, and the fourth number represents the left edge.

Submission Instructions. You must submit a ZIP file that includes **an output file** containing your best result and **a folder with your program**. Name your ZIP file exactly "Assignment1". ONLY SUBMIT ONE ZIP FILE PER TEAM.

The output file needs to be a .txt file with the names and IDs of all team members on the first line followed by your **best solution** (8 pieces per line, same as the input file). Make sure to not have empty spaces at the end of each line or an empty line at the end of the file. The only difference between the input file and the output file is the extra line at the beginning with your names and IDs (for a total of 9 lines). The file needs to be titled exactly "Ass1Output". Only submit one output file. I will pass the output file through the code posted on Moodle. **In your program, you must create a command line UI for me to input population size (in [100, 1000]) and number of generations (in [1, 100]); I will randomly choose two values to test your code. Besides, you need to make sure that the output file can be read by the code, or your assignment will not be graded.**

The program must be in Python, Java, or C++. The code in the **testing folder** simply reads your output file and counts the number of mismatching edges. The only thing you need to change in the file is the string file path. To test your solution in C++, create a new project and add the cpp to your project. Replace "/Users/zy/Documents/THOUGHT/Lab/PhD/Docs/TA folder/COEN 432 - FALL 23/Assignment#1/Testing/Ass1Output.txt" on line 16 by the path of your output file. To test your solution in Java, create a project with a package named Ass1 and place the file under the package. Replace "/Users/zy/Documents/THOUGHT/Lab/PhD/Docs/TA folder/COEN 432 - FALL 23/Assignment#1/Testing/Ass1Output.txt" on line 38 by the path of your output file.

Place the names and IDs of all team members on the first line of each file (commented). The program must read an input file named "Ass1Input", try to solve the 8x8 puzzle and produce an output file in the format mentioned above. If the submission is in C++, you must submit your cpp and header files. Include them in a single folder named "Ass1C++". If the submission is in Java, include all your .java files in a folder called "Ass1Java". If the submission is in Python, include all your .py files in a folder called "Ass1Python".

Marking Scheme. If a program does not run then it will not be marked (i.e., you will receive 0).

	Excellent (100%)	Good (80%)	Satisfactory (60%)	Unsatisfactory (<40%)
Efficiency (relative to other submissions) 30 points	Very efficient	Efficient	Slow	Extremely Slow
In-line Documentation 20 points	Very clear	Sufficient	Unclear	Missing
Correctness 50 points	0-15 Edge mismatches	16 - 30 mismatches	31 - 70 mismatches	>70 mismatches

Timeliness. Up to 24 hours of delay in submission leads to a 20% discount of your mark; > 24 hours, and your assignment may not be marked at all.

To grade the program, I will run a second input file and see if your program can produce an output that improves the solution. I will also look at the way you get to the solution (mutation, crossover, selection, etc.). For the output file, I will compare your results with those of your classmates. **Make sure your output file is in the same format as the example output file** – first line: names + ids, second line: 8 by 8 result.

You can use the online libraries if you need, here are some resources you can check out: LEAP(<https://leap-gmu.readthedocs.io/en/latest/>), DEAP(<https://deap.readthedocs.io/en/master/>), EO(<https://eodev.sourceforge.net/>).

For *clarifications* of the content of the assignment or submission procedure, e-mail the TA (zhiyangdeng.30@gmail.com)