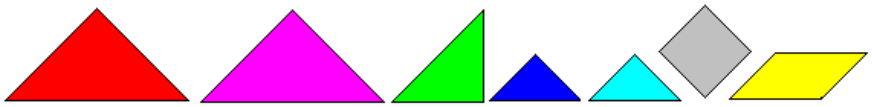
CS105
Fundamentals of Artificial Intelligence

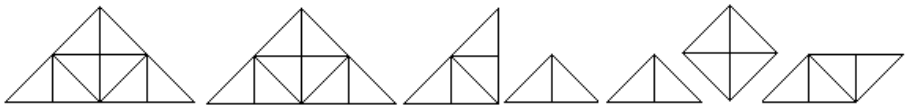
Group-project for Tangram Pieces Matching and Recognition

Tangram is one of the most popular games to play with. You put figures of 7 pieces together (five triangles, one square and one parallelogram). **You must use all pieces**. **They must touch but not overlap.**



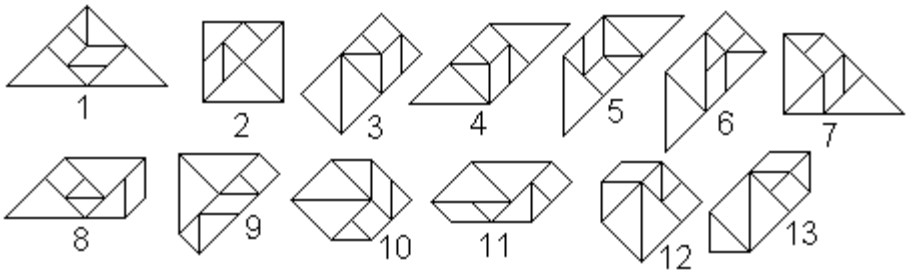
All seven tangram pieces consist of many half squares(triangles), each with this shape: 

There are 32 half squares or 16 squares altogether.



We could take the half-squared triangle as the basic form, because each square built for all seven tangram pieces has the simple length (perimeter) of 4 units.

There are only 13 convex figures that you can build from all the seven tangram pieces.



In this project, you are required to use AI search methods (uninformed search methods such as Breadth First search, Depth First search, Iterative Deepening and Uniform Cost search) and (Informed or heuristic search methods such as Greedy search, A* algorithm and Iterative Deepening version of A* : IDA*) to **perform pattern matching and recognition**. For some figures, there are more than one goal states (please see jpg files in the folder JPG in Moodle System) that could be reached. You are given all these 13 figures with no lines drawn inside each figure, i.e., they are figures with boundaries only (black box figures). Your jobs are given as follows:

1. Divide each figure into 32 half-squared (triangles) areas, please consult "Tangram Convex Figures" in the attachment in Moodle System.
2. Group these 32 areas into 7 larger areas: five triangles, one square and one parallelogram. The area ratios should be 8:8:4:2:2:4:4 =4:4:2:1:1:2:2 =LT:LT:MT:ST:ST:SQ:PA

3. Setup **important** problem formulation that is most suitable for solving this problem.
4. Use at least TWO uninformed and at least TWO informed search methods to solve this problem.
5. Implement these search methods in a computer system so that all the 7 pieces could be used to form 13 convex figures and the analyses of these methods could be done.
6. Analyze the properties of these search methods in terms of completeness, optimality, time complexity and space complexity when solving this problem.
7. You are required to form groups or team yourself. Each group should have 3 students. You are advised to include in your team/group at least ONE student who knows how to do programming, any programming languages, Matlab, Python, R, C, C++ and/or Java etc. could be used.
8. Submit to me student names, ID and class (e.g., D1 or D2) of your team members in week 5 in the class session. Students forming a team should come from the same class (D1 or D2).
9. A user friendly interface should be provided. A system name/title is given, e.g, "A Pattern Matching and Recognition System". The system should have options such as pull-down menu of 13 figures, pull-down menu of different search methods, etc. for user to choose from and execute. The results need to be displayed on the screen.
10. Each group or team is required to submit a report which includes the introduction, detailed description of your method(s), number of goal states (figures) you could reach, get or form, problems you have solved, your contributions, what you have achieved in your system.
11. A clear, succinct, easy to understand user manual/guide on how to INSTALL and EXECUTE the software (freeware or others) and your DEVELOPED system should be included in the report.
12. You are also required to submit a **full** and **detailed** report including the following information.
    - All your team members' name, ID number and team/group number (assigned in the week 12) and class (e.g., D1 or D2).
    - The problems you have solved.
    - Which uninformed or informed search methods you have used, how and why?
    - What evaluation function you have used to guide you in branching (expanding) nodes?
    - How and why such functions or changes are introduced? What are the reasons behind introducing such changes?
    - Figures or diagrams showing the problem formulation of your system should be included.
    - A comparison of properties of different adopted or implemented search methods in terms of Completeness, Optimality, Time and Space complexity need to be done and included.
    - The results you have obtained. Different figures showing results, many goal states, etc.
13. You are required to submit in the last/final class session your **Hardcopy** report and a **USB** which contains the source codes of your developed system, shareware and/or freeware used, user manual/guide on how to INSTALL and EXECUTE your system, and your **FULL report** mentioned in points 10 -- 12.