Games and Computation Homework #4: Heuristic Evaluation and Alpha-Beta Pruning

Answer these questions within the HW #4 Moodle quiz:

Heuristic / Static Evaluation Function

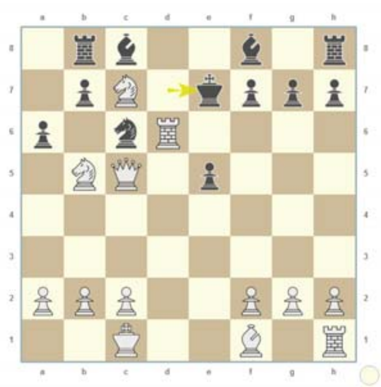
When we cannot evaluate the entire game-tree/-graph, one approach is to develop a computationally efficient means of *estimating* the value of any given node and using it to evaluate non-terminal nodes. One can then perform minimax using these values. We call the function that computes such an estimate a *heuristic* or *static evaluation function*.

For the game of Chess, one of the simplest means of heuristic evaluation is to assign a positive or negative value to each non-king piece of Max (White) or Min (Black), respectively, and sum the material value on the board. The Fruit chess engine (2005) assigns the following absolute values to each of the non-king pieces:

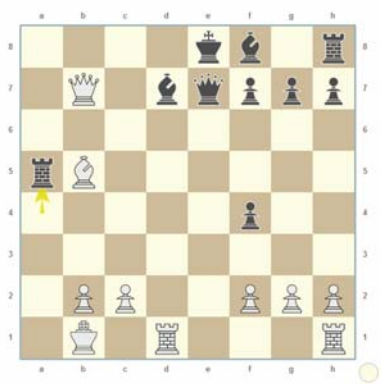
- Pawn = 100
- Knight = 400
- Bishop = 400
- Rook = 600
- Queen = 1200

For each of the following boards, assume that first-player White is the maximizing player and second-player Black is the minimizing player. Evaluate each of the following boards according to Fruit's material heuristic evaluation function:

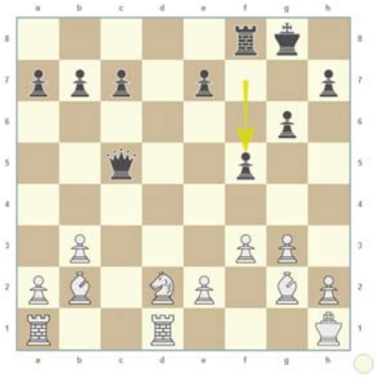
Board 1 Value: ________



Board 2 Value: ________



Board 3 Value: ________



FreeCell Features for Heuristic Evaluation Functions

A *feature* of a game state is a measurable characteristic that is generally used to construct or learn heuristic evaluation functions. For Chess, evaluation features include: material value (as in the previous exercises), mobility (often measured as the number of legal moves), piece-square tables (assigning values to pieces being in certain positions), pawn structure scores, etc.

For this essay question, play and study the strategy of the solitaire card game FreeCell (available on our system under Games → AisleRiot Solitaire). Then describe three state features that might serve as measures (e.g. beginning with "the number of…"/"the count of …") used to evaluate a FreeCell board with higher evaluations indicating a higher probability of solving the game through that state:

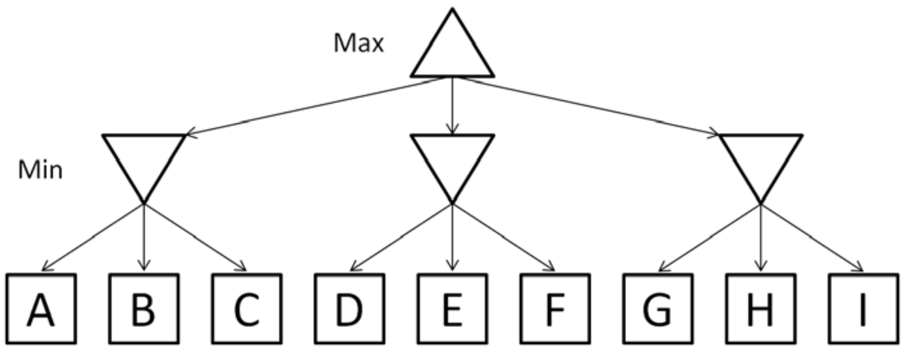
Feature 1: ____

Feature 2: ____

Feature 3: ____

Alpha-Beta Pruning

Consider the following Minimax Tree. If we evaluate children from left-to-right using alpha-beta pruning with a given set of values for nodes A through I, some of the nodes may be pruned. That is, we may never need to evaluate the node because of what we have seen in prior search.



For the following values of nodes A through I, enter the only the letters of the pruned nodes in alphabetical order with *no spaces or commas in between*.

Minimax Tree #1: `A=5  B=3  C=1  D=2  E=5  F=4  G=1  H=3  I=3`

Pruned nodes? _________

Minimax Tree #2: `A=5  B=2  C=2  D=5  E=1  F=3  G=2  H=4  I=2`

Pruned nodes? _________

Minimax Tree #3: `A=1  B=3  C=4  D=1  E=4  F=1  G=3  H=5  I=3`

Pruned nodes? _________