Guidance to Solving the Nimgame Programming Project

(1) Dumb version

(1.1) Goal: either the computer or the opponent can win.

(1.2) strategy: the computer uses the random number generator to pick up some coins (between 1 and *k* (inclusively) and less than or equal to *m*, where *k* is the number of coins left and *m* is the maximum number of coins the computer and the opponent can pick up each time).

(2) Smart version

(2.1) Goal: the computer tries its best to win

(2.2) Option 1: the one picking up the last coin(s) wins

(2.2.1) Strategy: if $k \% (m + 1) \neq 0$ and it's the opponent's turn, then the opponent is currently in the winning state. What the opponent needs to do in order to stay in the winning state is to pick up the number of coins (e.g. *c*) that will make the following expression *true*: $(k - c) \% (m + 1) = 0$. This guarantees to leave the computer in the losing state.

(2.2.2) An example: if $k = 8$, $m = 3$, and it's the opponent's turn, then the opponent will lose because $8 \% (3 + 1) = 0$ meaning that the computer has just made a smart move for its turn that puts its opponent in a losing state. The computer will eventually win the game if it keeps moving from the winning state to another winning state.

(a) Suppose the opponent picks up 2 coins, then $k = 6$ now.
(b) Suppose the computer then picks up 2 coins, then $k = 4$ and $4 \% (3 + 1) = 0$, which continues to put its opponent in another losing state. However, if the computer picks up 1 coin, then $5 \% (3 + 1) \neq 0$, which will put its opponent in the winning state. Therefore, the computer will pick up 2 coins and $k = 4$ now.

(c) If the opponent now picks up 1 coin, then the computer wins because it can pick up all the 3 coins remaining. If the opponent picks up 2 coins, then the computer still wins because it can pick up the last 2 coins. If the opponent picks up 3 coins, then the computer still wins because it can pick up the last coin.

(2.3) Option 2: the one picking up the last coin(s) loses

(2.3.1) Strategy: if $(k - 1) \% (m + 1) \neq 0$ and it's the opponent's turn, then the opponent is currently in the winning state. What the opponent needs to do in order to stay in the winning state is to pick up the number of coins (e.g. c) that will make the following expression *true*: $(k - 1 - c) \% (m + 1) = 0$. This guarantees to leave the computer in the losing state.

(2.3.2) The same example: $k = 8$, $m = 3$, and it's the opponent's turn, then the opponent will win because $(8 - 1) \% (3 + 1) \neq 0$ meaning that the computer did NOT make a smart move to put its opponent in a losing state. The opponent will eventually win the game if she/he keeps moving from the winning state to another winning state.

(a) If the opponent picks up 1 coin, then it is NOT a smart move (why?) and will put the computer in the winning state.
(b) If the opponent picks up 2 coins, then it is NOT a smart move (why?) and will put the computer in the winning state.
(c) If the opponent picks up 3 coins, then it is a smart move (why?) and will put the computer in the losing state.
(d) So the opponent determines to pick up 3 coins and $k = 5$.
(e) Now, it's the computer's turn and suppose the computer picks up 1 coin, then the opponent will win because $k = 4$ and the opponent can pick up 3 coins to leave the last coin to the computer. Suppose the computer picks up 2 coins, then the opponent still wins because $k = 3$ and the opponent can pick up 2 coins to leave the last coin to the computer. Suppose the computer picks up 3 coins, then the opponent still wins because $k = 2$ and the opponent can pick up 1 coin to leave the last coin to the computer. In other words, once the computer is currently

in the losing state, it will use the random number generator to determine the number of coins to pick up because it really does NOT matter how many coins the computer should pick up in order not to lose the game.

(3) Coach version

(3.1) Goal: the opponent is under the guidance of the computer to learn how to win the game

(3.2) Strategy: the computer always advises the opponent to keep staying in the winning state guided by the formulas shown above. In other words, the computer should make a wrong move ON PURPOSE from the winning state to the losing state to give the opponent a chance to learn how to win the game. However, if the opponent is currently in the winning state and is about to make a move from the winning state to the losing state, the computer should always give some advice to the opponent to guide him/her regarding the potential mistake he/she is about to make on the move and therefore learn how to win the game. The advice made by the computer might be just a message such as "Are you sure you want to pick up 2 coins?"