Matthew Stroble  
Bonus 2

Had to put code on different lines for readability

 current\_highest\_value = self.max\_score()

        board\_sum = self.sum\_board()

        average = float(board\_sum) / (self.height \* self.width)

        if algoSelection == 1:

            num\_of\_zero = (self.height \* self.width) - self.count\_nonzero()

            moves\_left = ((self.Goal - current\_highest\_value))

            penalty = self.count\_distinct()

            # non consistant admissible heuristic

            heuristic = moves\_left \* 3 + penalty –

num\_of\_zero - (float(self.Goal) / average) + len(self.moveHistory)

This heuristic takes a combination of things and is admissible I as it never goes negative and is not consistent since some children generate higher heuristic values than the parent. Essentially what it does is penalize the board that has the most amount of distinct values and favors boards with more combined values. Not sure how to prove this one.