**Homework 2: N-Queens, hill climb (2+B)**

Submitted by Amit Ben Zaken

1. View the .ipynb file for the code implementation.

We have the main class NQueens which has self.N and self.board. N is the size of the board and the board is stored as a list of size N where the i’th entry represents the row of the queen in the i’th column.

1. Purpose function – count\_attacks()

count\_attacks() is the function that computes the number of pairs of queens attacking each other (even if there is a queen intercepting it counts as an attack).

The goal is to minimize this function until it reaches 0, meaning no queens are attacking each other, that will be a valid solution. It is worth mentioning there is a valid solution for all N>=4.

1. The environment consists of all possible board states that differ from the current board by moving a single queen to a different row in its column.

This is implemented in the get\_neighbors() function.

1. Our method for enumerating states is by generating all neighbors of the current state. Every iteration we choose the state with the minimal attacks.
2. There is an implementation of a random restart in the case that no neighbor improves the number of attacks (i.e. stuck on a local minimum).