LP4: SAT Solver using BEE

Kakuro:
- **Verify:** We assume that the hints and variables of the instance and solution are ordered in the same manner.
- **Encode**: When creating the map, we first flattened the lists of the variables into one list. After that, we removed the duplicates. We mapped each hint into a list of [HINT_VALUE=[LIST OF VARIABLES]], where the variables are the row/column that are summed in order to get the HINT_VALUE that is given in the instance.
  The **constraints** we used:
    - new_int in order to create numbers from 1 to 9 (including).
    - int_array_allDiff: For each value, the variables that sum up to it all must be distinct which is why we used this constraint on the list of variables in the map described above.
    - Int_array_plus: The sum of the list of variables for each hint must equal the hint. This constraint ensures that.

Scheduling:
- **Encode**: We mapped the instance with a list of size |Nexams|. Each index in the list represents the exam as it is resembled in the conflicts list. The values in the list are the time slots that the exam is scheduled to. This is O(NExams) because in the worst case all of the exams will conflict in pairs and therefore the output will be a list of size Nexams with the max value equal to |Nexams| and the min value equal to 1.
  The **constraints** we used:
    - Made sure that for every conflict c(I,J), I < J.
    - new_int in order to create the list timeslots (our map).
    - Int_neq in order to make sure that for every conflict c(I,J), the value at timeslots[I] is not equal to the value at timeslots[J].

We, Arnon Kahani (ID 305509556) and Niv Lipetz (ID 305214801), assert that the work we submitted is entirely or own. We have not received any part from any other student in the class (or other source), nor did we give parts of it for use to others. We have clearly marked in the comments of our program any code taken from an external source. We assert that the division of work between us is: Arnon Kahani 50% and Niv Lipetz 50%.