KENKEN Puzzle

Course Code: **CSE481**

Course Name: **Artificial Intelligence**

Ain Shams University
Faculty of Engineering
Spring Semester – 2022

Student Personal Information

عمرو ايهاب عبدالعزيز محمد عبدالعزيز	1700923
عمر محمد عمر عبدالسميع	1700903
محمد خالد سيد ابراهيم مهران	1601166
عمرو احمد محمد فتحى محمد	1700918
لؤي عبدالله يوسف عبدالهادي	1701043
عمر محمد محمد مصطفى	1700907

GitHub Repo Link

https://github.com/AmrAhmed11/kenken-puzzle-solver

Performance Analysis

Total Backtracking Search: 4.9437 Seconds

Total Backtracking Search with Arc Consistency: 0.3650 Seconds

Total Backtracking Search with Forward Checking: 0.0533 Seconds

Average Backtracking Search Per Test: 0.0494 Seconds

Average Backtracking Search with Arc Consistency Per Test: 0.0036 Seconds

Average Backtracking Search with Forward Checking Per Test: 0.0005 Seconds

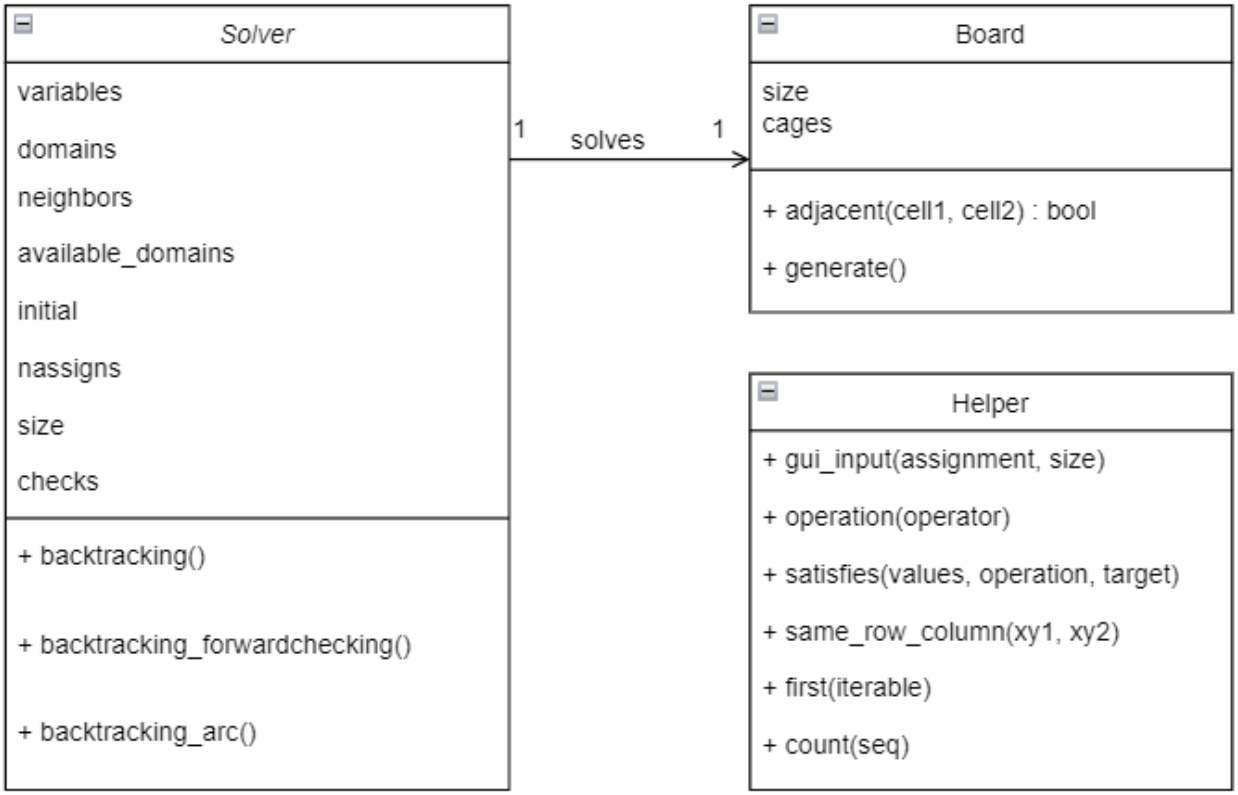
Average Grid Size: 4

Re-run Tests Run Game

We run the three algorithms on 100 randomly generated boards with random sizes from 2 to 6.
The speed ranks:

1- Backtracking search with forward checking.
2- Backtracking search with arc consistency.
3- Backtracking search.

Class diagram



Data structures

Board is represented as a list of tuples. Each tuple represents a cage. The cage tuple has 3 fields

The first one is a tuple of tuples. Each one represents the coordinates of a cell that belongs to the cage.

The second field is a character that represents the operation used in the cage (+, -, *, /, .)

The third field is an integer that represents the target result of the cage.

```
[(((1, 1),), '.', 2), (((2, 1), (3, 1)), '/', 3), (((1, 2), (1, 3)), '-', -2), (((2, 2), (2, 3), (3, 3), (3, 2)), '+', 8)]
```

An **Assignment** is represented as a dictionary, there is a key for each cage on the board.

The key is a tuple of tuples, where each of them is a coordinate of a cell in the cage.

The value is a tuple of the values of each cell in the cage.

```
{((1, 1),): (2,), ((2, 1), (3, 1)): (1, 3), ((1, 2), (1, 3)): (1, 3), ((2, 2), (2, 3), (3, 3), (3, 2)): (3, 2, 1, 2)}
```