**VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY**

**INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**Ảnh có chứa vòng tròn, biểu tượng, Nhãn hiệu, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

**ARTIFICIAL INTELLIGENCE**

**IT159IU**

**REPORT LAB 4-5**

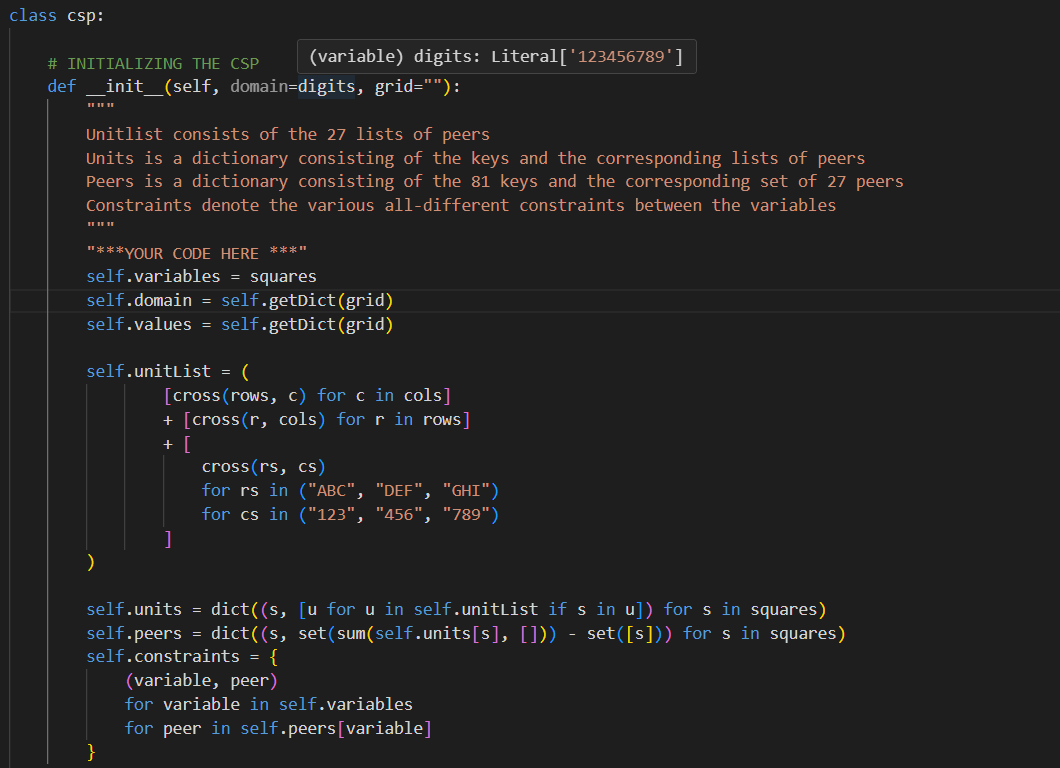
**Instructor:**

**Dr. Nguyen Trung Ky**

**Dr. Ly Tu Nga**

**Nguyen Huynh Ngan Anh - ITDSIU23003**

1. **Exercise 1:**

****

**Key Components of the CSP Representation**

* Variables:
  + Represented by [self.variables](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), which is a list of all 81 cells in the Sudoku grid (e.g., "A1", "A2", ..., "I9").
  + These variables correspond to the rows (A-I) and columns (1-9) of the grid.
* Domains:
  + Represented by [self.domain](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [self.values](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), which are dictionaries mapping each variable to its possible values.
  + Initially, each variable's domain is "123456789" unless the grid specifies a pre-filled value.
* Units:
  + Represented by [self.unitList](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), which contains 27 lists:
    - 9 rows
    - 9 columns
    - 9 sub-grids (3x3 blocks)
  + Each unit is a group of variables that must satisfy the "all-different" constraint.
* Peers:
  + Represented by [self.peers](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), which is a dictionary mapping each variable to the set of other variables in the same row, column, or sub-grid.
  + Peers are used to enforce constraints during the solving process.
* Constraints: represented by [self.constraints](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), which is a set of all variable-peer pairs that must satisfy the "all-different" constraint.

**Design Decisions**

* Grid Representation:
  + The grid is represented as a single string of 81 characters, where each character corresponds to a cell in the Sudoku grid.
  + The [getDict](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method converts this string into a dictionary mapping variables to their values or possible domains.
* Units and Peers:
  + Units are precomputed to simplify constraint checking.
  + Peers are derived from units, ensuring that each variable has a direct reference to its related variables.
* Constraints: constraints are implicitly enforced using the [peers](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) dictionary, which avoids the need to explicitly store all constraints.

**How the Implementation Works**

* Initialization ([\_\_init\_\_](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)):
  + The [\_\_init\_\_](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method initializes the CSP by defining variables, domains, units, peers, and constraints.
  + It uses helper functions like [cross](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to generate units and peers.
* Domain Initialization ([getDict](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)): the [getDict](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method processes the input grid string and assigns values to variables:
  + - If a cell is pre-filled (non-zero), its domain is restricted to that value.
    - Otherwise, the domain is set to "123456789".
* Units and Peers:
  + Units are generated for rows, columns, and sub-grids using the [cross](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) function.
  + Peers are computed by summing all units containing a variable and removing the variable itself.

**I choose this design because:**

* **Efficiency**: precomputing units and peers reduces the computational overhead during constraint checking.
* **Modularity**: the design separates variables, domains, units, peers, and constraints, making the code easier to understand and extend.
* **Scalability**: the representation can be adapted to other CSP problems by modifying the variables, domains, and constraints.
* **Alignment with CSP Principles**: the implementation adheres to the standard CSP framework, making it compatible with generic CSP-solving algorithms like backtracking and forward checking.

1. **Exercise 2:**

Implement backtracking search algorithm

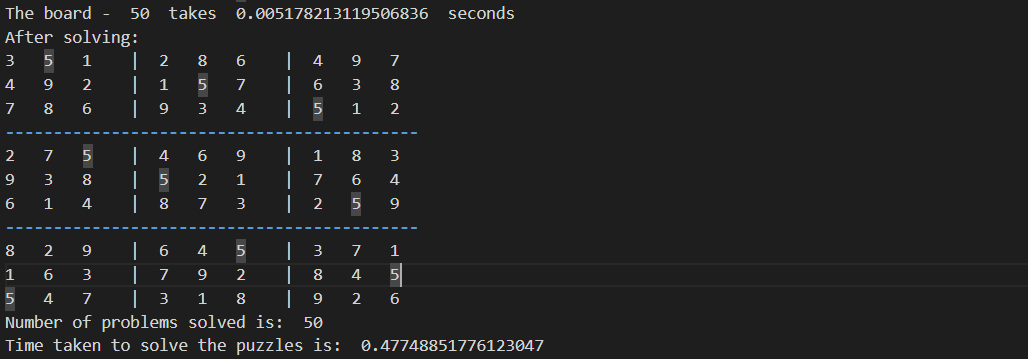
A black screen with white text

AI-generated content may be incorrect.

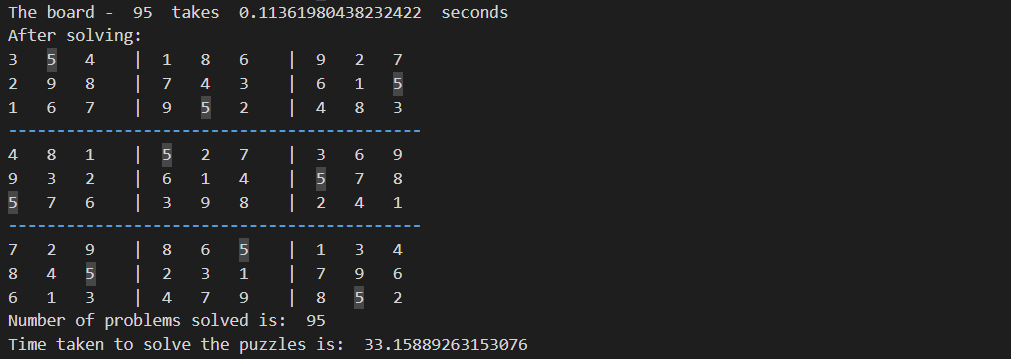
A screen shot of a computer program

AI-generated content may be incorrect.

Test the backtracking search for euler puzzles



Test the backtracking search for magictour puzzles



1. **Exercise 3:**

Implement the AC3 algorithm

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

Change the algorithm used in sudoku.pyA screen shot of a computer program

AI-generated content may be incorrect.

Test the backtracking search with AC3 algortihm for preprocessing for euler puzzles

A screen shot of a computer

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

Test the backtracking search with AC3 algortihm for preprocessing for magictour puzzles

A screen shot of a computer

AI-generated content may be incorrect.

1. **Experience description:**

It is a fascinating application of AI techniques. I am very interesting to see how the design options of variables, domains, and constraints can be used to represent the Sudoku puzzle as a CSP, and how the backtracking algorithm can be used to efficiently search for solutions. I find it a difficulty that can arise is choosing the right heuristics for variable and value selection, which can significantly affect the efficiency of the solution process. Overall, it was a fun and challenging assignment that showcases the power of AI in solving complex problems. In addition, the Assignment is difficult. When I initially looked at the specifications and the source code offered, I was confused. Just to get the gist of what to accomplish, I have to browse through theoretical presentations and web resources. I finally finish it after working with the lab together and using several examples

1. **Time spend:** it took me around 4 hours to complete the lab