CSC361: Artificial Intelligence

Assignment #2

Student Name	Student ID	Section#	Tasks
Norah Alguraishi	442201375	44039	The work on this assignment was shared among students and done while meeting together
Maha Albawardy	442200383		
Sarah Alajlan	442202314		

Simple Backtracking

The `SimpleBacktracking` class implements the simple backtracking algorithm. It receives the two dimensional randomly generated grid with some empty cells in the `SimpleBacktrackingSearch` method and solves it first by initializing various variables such as `NoVarAssigend` which is an integer to store the number of variable assignments, `NoConChecks` which is an integer that stores the number of consistency checks, `startTime` which is a double that stores the start time of the search, and `nextSelections` which is an array list of pairs of the location of the empty cells. It loops through all the cells and adds the unassigned cells to `nextSelections` list. After that it calls the recursive boolean method `SimpleBacktrackingRec` that performs the simple backtracking by checking if assignments are complete it returns true, if they're not complete, it gets the next unassigned cell and loops through the domain, assigning values to the cell and checking if the assignment is consistent with the constraints by calling the `checkConstraint` method in the `FunctionsClass` class.

Backtracking with MRV heuristic

In this class we have :
`NoVarAssigned`: a counter that keeps track of the number of variables assigned a value, `NoConChecks`: a counter that keeps track of the number of constraint checks performed, `startTime`: the start time of the algorithm, used to calculate the total time taken Also the class has two methods: `BacktrackingMRV`: the main method of the class, which initializes the counters and starts the backtracking process `BacktrackingMRVRec`: the recursive backtracking method, which implements the MRV heuristic. The BacktrackingMRVRec method has the following steps: Check if the puzzle is complete, and if so, return true
Identify the location with the minimum number of remaining values (using the MRV heuristic). Try all possible values for the selected location and perform constraint checks If a value results in a valid solution, return true, otherwise

continue trying different values If all values have been tried and none results in a valid solution, return false The CountConstraints method is a helper method used to count the number of constraints for a given cell location.

Forward checking

`NoVarAssigend`: an integer to keep track of the number of assigned variables (i.e., cells with a value in the puzzle)
`NoConChecks`: an integer to keep track of the number of constraint checks performed `startTime`: a double to keep track of the starting time of the search `nextSelections`: a list of lists of arrays to keep track of the next possible selections (i.e., the next unassigned cell and its domain)
The main function of the class is `ForwardCheckingSearch`, which takes as input a two-dimensional integer array grid representing the initial puzzle. The function performs the following steps: Initializes the variables `NoVarAssigend`, `NoConChecks`, and `startTime` Creates a list `nextSelections` to store the next possible selections Loops over all cells of the puzzle and adds an unassigned cell to nextSelections Calls the recursive function `ForwardCheckingRec` to start the search Prints the final puzzle and some information about the search The function `ForwardCheckingRec` performs the forward checking algorithm. It takes as input the current puzzle and does the following: If the puzzle is complete (i.e., all cells are assigned), it returns true Gets the next unassigned cell n from nextSelections Loops over all values from 0 to 9 and tries to assign the value to the cell If the assigned value is not in the domain of the cell, it continues to the next value If the assigned value is in the domain of the cell, it updates the puzzle and calls ForwardCheckingRec recursively. If the recursive call returns false, it removes the assignment and adds the cell back to nextSelectionsIf all values have been tried and none of them leads to a complete puzzle, it returns falseThe class also has several helper functions, such as forwardCheck and searchDomain, which are used to update the domains of the cells during the search.

Sample runs and analysis

Sample run1 :

```
The initial state is:
========================
   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   9  |   0  |   1  |
   -1  |   -1  |   0  |   7  |   -1  |   -1  |   6  |   8  |   3  |   4  |
   -1  |   7  |   -1  |   1  |   8  |   -1  |   2  |   0  |   5  |   6  |
   15  |   13  |   11  |   14  |   20  |   11  |   15  |   17  |   8  |   11  |

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 1
========================
simple backtrack search:
========================

   4  |   5  |   8  |   6  |   3  |   2  |   7  |   9  |   0  |   1  |
   2  |   1  |   0  |   7  |   9  |   5  |   6  |   8  |   3  |   4  |
   9  |   7  |   3  |   1  |   8  |   4  |   2  |   0  |   5  |   6  |
   15  |   13  |   11  |   14  |   20  |   11  |   15  |   17  |   8  |   11  |
the number of variable assignments is :5352
the number of consistincy checks is :7398
The total time is :11.0

========================
```

```
========================
forward check
========================

   4  |   5  |   8  |   6  |   3  |   2  |   7  |   9  |   0  |   1  |
   2  |   1  |   0  |   7  |   9  |   5  |   6  |   8  |   3  |   4  |
   9  |   7  |   3  |   1  |   8  |   4  |   2  |   0  |   5  |   6  |
   15  |   13  |   11  |   14  |   20  |   11  |   15  |   17  |   8  |   11  |
the number of variable assignments is :3188
the number of consistincy checks is :6112
The total time is :19.0

========================


1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 3
========================
backtrack with mrv heuristic
========================
   4  |   5  |   8  |   6  |   3  |   2  |   7  |   9  |   0  |   1  |
   2  |   1  |   0  |   7  |   9  |   5  |   6  |   8  |   3  |   4  |
   9  |   7  |   3  |   1  |   8  |   4  |   2  |   0  |   5  |   6  |
   15  |   13  |   11  |   14  |   20  |   11  |   15  |   17  |   8  |   11  |
the number of variable assignments is :92
the number of consistincy checks is :140
The total time is :5.0

========================
```

Sample run2 :

```
The initial state is:
========================
   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   3  |   4  |
   -1  |   -1  |   1  |   -1  |   0  |   5  |   3  |   -1  |   7  |   9  |
   1  |   -1  |   0  |   9  |   7  |   -1  |   8  |   5  |   3  |   6  |
   13  |   10  |   6  |   26  |   14  |   15  |   12  |   7  |   13  |   19  |

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 1
========================
simple backtrack search:
========================

   6  |   2  |   5  |   9  |   7  |   8  |   1  |   0  |   3  |   4  |
   6  |   4  |   1  |   8  |   0  |   5  |   3  |   2  |   7  |   9  |
   1  |   4  |   0  |   9  |   7  |   2  |   8  |   5  |   3  |   6  |
   13  |   10  |   6  |   26  |   14  |   15  |   12  |   7  |   13  |   19  |
the number of variable assignments is :3418
the number of consistincy checks is :4853
The total time is :8.0

========================
```

```
========================
forward check
========================

   8  |   2  |   5  |   9  |   7  |   6  |   1  |   0  |   3  |   4  |
   4  |   6  |   1  |   8  |   0  |   5  |   3  |   2  |   7  |   9  |
   1  |   2  |   0  |   9  |   7  |   4  |   8  |   5  |   3  |   6  |
   13  |   10  |   6  |   26  |   14  |   15  |   12  |   7  |   13  |   19  |
the number of variable assignments is :2648
the number of consistincy checks is :5186
The total time is :18.0

========================

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 3
========================
backtrack with mrv heuristic
========================
   8  |   2  |   5  |   9  |   7  |   6  |   1  |   0  |   3  |   4  |
   4  |   6  |   1  |   8  |   0  |   5  |   3  |   2  |   7  |   9  |
   1  |   2  |   0  |   9  |   7  |   4  |   8  |   5  |   3  |   6  |
   13  |   10  |   6  |   26  |   14  |   15  |   12  |   7  |   13  |   19  |
the number of variable assignments is :78
the number of consistincy checks is :116
The total time is :4.0

========================
```

Sample run3 :

```
The initial state is:
========================
   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   9  |   -1  |
   -1  |   -1  |   0  |   9  |   -1  |   -1  |   5  |   -1  |   1  |   2  |
   0  |   2  |   4  |   8  |   1  |   9  |   7  |   6  |   5  |   3  |
   12  |   12  |   7  |   23  |   5  |   23  |   13  |   16  |   15  |   9  |

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 1
========================
simple backtrack search:
========================

   5  |   2  |   3  |   6  |   0  |   8  |   1  |   7  |   9  |   4  |
   7  |   8  |   0  |   9  |   4  |   6  |   5  |   3  |   1  |   2  |
   0  |   2  |   4  |   8  |   1  |   9  |   7  |   6  |   5  |   3  |
   12  |   12  |   7  |   23  |   5  |   23  |   13  |   16  |   15  |   9  |
the number of variable assignments is :4768
the number of consistincy checks is :6939
The total time is :13.0

========================
```

```
========================
forward check
========================

   5  |   2  |   3  |   6  |   0  |   8  |   1  |   7  |   9  |   4  |
   7  |   8  |   0  |   9  |   4  |   6  |   5  |   3  |   1  |   2  |
   0  |   2  |   4  |   8  |   1  |   9  |   7  |   6  |   5  |   3  |
   12  |   12  |   7  |   23  |   5  |   23  |   13  |   16  |   15  |   9  |
the number of variable assignments is :3287
the number of consistincy checks is :7065
The total time is :23.0

========================

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 3
========================
backtrack with mrv heuristic
========================
   5  |   2  |   3  |   6  |   0  |   8  |   1  |   7  |   9  |   4  |
   7  |   8  |   0  |   9  |   4  |   6  |   5  |   3  |   1  |   2  |
   0  |   2  |   4  |   8  |   1  |   9  |   7  |   6  |   5  |   3  |
   12  |   12  |   7  |   23  |   5  |   23  |   13  |   16  |   15  |   9  |
the number of variable assignments is :138
the number of consistincy checks is :212
The total time is :7.0

========================
```

Sample run4 :

```
The initial state is:
========================
   -1  |   -1  |   -1  |   -1  |   -1  |   -1  |   7  |   -1  |   9  |   8  |
   -1  |   -1  |   4  |   5  |   -1  |   1  |   -1  |   8  |   2  |   -1  |
   -1  |   0  |   -1  |   2  |   7  |   4  |   5  |   1  |   9  |   8  |
   13  |   8  |   16  |   9  |   16  |   8  |   12  |   14  |   20  |   19  |

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 1
========================
simple backtrack search:
========================

   4  |   1  |   6  |   2  |   0  |   3  |   7  |   5  |   9  |   8  |
   6  |   7  |   4  |   5  |   9  |   1  |   0  |   8  |   2  |   3  |
   3  |   0  |   6  |   2  |   7  |   4  |   5  |   1  |   9  |   8  |
   13  |   8  |   16  |   9  |   16  |   8  |   12  |   14  |   20  |   19  |
the number of variable assignments is :3249
the number of consistincy checks is :4724
The total time is :10.0

========================
```

```
========================
forward check
========================

   4  |   1  |   6  |   2  |   0  |   3  |   7  |   5  |   9  |   8  |
   6  |   7  |   4  |   5  |   9  |   1  |   0  |   8  |   2  |   3  |
   3  |   0  |   6  |   2  |   7  |   4  |   5  |   1  |   9  |   8  |
   13  |   8  |   16  |   9  |   16  |   8  |   12  |   14  |   20  |   19  |
the number of variable assignments is :2401
the number of consistincy checks is :4418
The total time is :15.0

========================

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 3
========================
backtrack with mrv heuristic
========================
   4  |   1  |   6  |   2  |   0  |   3  |   7  |   5  |   9  |   8  |
   6  |   7  |   4  |   5  |   9  |   1  |   0  |   8  |   2  |   3  |
   3  |   0  |   6  |   2  |   7  |   4  |   5  |   1  |   9  |   8  |
   13  |   8  |   16  |   9  |   16  |   8  |   12  |   14  |   20  |   19  |
the number of variable assignments is :79
the number of consistincy checks is :121
The total time is :6.0
```

Sample run5 :

```
The initial state is:
========================
   -1  |   -1  |   -1  |   -1  |   -1  |   9  |   -1  |   1  |   -1  |   5  |
   -1  |   -1  |   5  |   -1  |   -1  |   1  |   6  |   3  |   -1  |   8  |
   -1  |   3  |   0  |   4  |   8  |   5  |   7  |   2  |   6  |   -1  |
   10  |   15  |   13  |   12  |   12  |   15  |   20  |   6  |   10  |   22  |

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 1
========================
simple backtrack search:
========================

   2  |   3  |   8  |   6  |   4  |   9  |   7  |   1  |   0  |   5  |
   7  |   9  |   5  |   2  |   0  |   1  |   6  |   3  |   4  |   8  |
   1  |   3  |   0  |   4  |   8  |   5  |   7  |   2  |   6  |   9  |
   10  |   15  |   13  |   12  |   12  |   15  |   20  |   6  |   10  |   22  |
the number of variable assignments is :2016
the number of consistincy checks is :2832
The total time is :10.0

========================
```

```
========================
forward check
========================

   2  |   3  |   8  |   6  |   4  |   9  |   7  |   1  |   0  |   5  |
   7  |   9  |   5  |   2  |   0  |   1  |   6  |   3  |   4  |   8  |
   1  |   3  |   0  |   4  |   8  |   5  |   7  |   2  |   6  |   9  |
   10  |   15  |   13  |   12  |   12  |   15  |   20  |   6  |   10  |   22  |
the number of variable assignments is :863
the number of consistincy checks is :1613
The total time is :10.0

========================

1- Simple backtracking
2- Backtracking with forward checking
3- Backtracking with MRV
4- Exit
Enter a choice: 3
========================
backtrack with mrv heuristic
========================
   2  |   3  |   8  |   6  |   4  |   9  |   7  |   1  |   0  |   5  |
   7  |   9  |   5  |   2  |   0  |   1  |   6  |   3  |   4  |   8  |
   1  |   3  |   0  |   4  |   8  |   5  |   7  |   2  |   6  |   9  |
   10  |   15  |   13  |   12  |   12  |   15  |   20  |   6  |   10  |   22  |
the number of variable assignments is :76
the number of consistincy checks is :123
The total time is :4.0
```

Simple Backtracking:
Average of Number of consistency checks is
(7398 + 4853 + 6939 + 4724 + 2832) / 5 = 5349.2

Backtracking with Forward checking:
Average of Number of consistency checks is
(6112 + 5186 + 7065 + 4418 + 1613) / 5 = 4878.8

Backtracking with MRV:
Average of Number of consistency checks is
(140 + 116 + 212 + 121 +123) / 5 = 142.4

Conclusion:
We did 5 runs on 5 different random 4 x 10 grids and we noticed that Backtracking with MRV was better in terms of having a lower number of variable assignment and consistency checks also it takes less time than the other two methods.