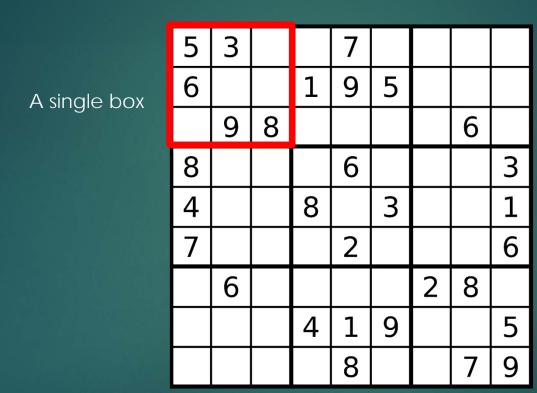
SUDOKU

BACKTRACKING

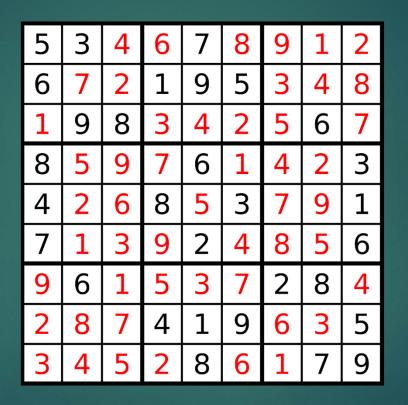


| 5 6 | 3 | | | 7 | | | | |
|--------|---|---|---|---|---|---|---|---|
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | | 6 | | | | 3 |
| 4 | | | 8 | | 3 | | | 1 |
| 7 | | | | 2 | | | | 6 |
| | 6 | | | | | 2 | 8 | |
| | | | 4 | 1 | 9 | | | 5 |
| | | | | 8 | | | 7 | 9 |



<u>Sudoku</u>

- ► The aim of sudoku is to fill a 9×9 chessboard-like grid with digits
- ▶ We have some <u>rules</u>:
- Each column + each row, and each of the nine 3×3 sub-grids that compose the grid (boxes) contains all of the digits from 1 to 9
- ▶ Initially we have → a partially completed grid, which for a wellposed puzzle has a unique solution
- ▶ The same integer may not appear twice in the same row + column or in any of the nine 3×3 subregions / boxes of the 9x9 grid



<u>Sudoku</u>

- ► The problem itself is NP-complete
- Running time complexity: O(mⁿ)
- ▶ m: number of possibilities for a single cell (9)
- n: number of blank fields at the beginning
- ▶ Backtracking:
- Iterates all the possible solutions for the given Sudoku
- ▶ If the solutions assigned do not lead to the solution of Sudoku, the algorithm discards the solutions and rollbacks to the original solutions and retries again

Search tree



