

Zespół Szkół NR.10  
Im. Stanisława Staszica

Sprawozdanie z projektu  
Algorytm do rozwiązywania Sudoku

Autor: Antoni Paczkowski  
Warszawa, 29.09.2025 r.

**1. Cel i Uzasadnienie Projektu**

Celem głównym projektu jest zaprojektowanie i implementacja wydajnej aplikacji w języku C#, zdolnej do automatycznego rozwiązywania łamigłówek Sudoku o standardowej siatce  $9 \times 9$ . Program ma przyjmować planszę wejściową (częściowo wypełnioną lub pustą) i generować unikatowe, poprawne rozwiązanie zgodne z regułami gry w możliwie najkrótszym czasie.

## **2. Opis Problemu i Kontekst**

Sudoku jest klasyczną łamigłówką logiczną polegającą na wypełnieniu siatki  $9 \times 9$  cyframi od 1 do 9. Plansza jest podzielona na dziewięć mniejszych bloków  $3 \times 3$ . Kluczowe reguły, które musi spełnić rozwiązanie, to:

Każda cyfra od 1 do 9 musi wystąpić dokładnie raz w każdym z 9 wierszy.

Każda cyfra od 1 do 9 musi wystąpić dokładnie raz w każdej z 9 kolumn.

Każda cyfra od 1 do 9 musi wystąpić dokładnie raz w każdym z 9 mniejszych bloków  $3 \times 3$ .

Ręczne rozwiązywanie Sudoku, szczególnie na wyższych poziomach trudności, jest procesem czasochłonnym i podatnym na błędy. Projekt ma na celu stworzenie narzędzia automatyzującego to rozumowanie.

## **3. Zakres Projektu Projekt zostanie zrealizowany w kilku kluczowych etapach:**

Wczytywanie Danych Wejściowych: Program musi być zdolny do przyjęcia planszy Sudoku  $9 \times 9$  od użytkownika.

Walidacja Poprawności Planszy: Sprawdzenie, czy wstępnie wprowadzone cyfry nie naruszają zasad Sudoku (unikalność w wierszach, kolumnach i blokach  $3 \times 3$ ).

Implementacja Algorytmu Rozwiązywającego: Zastosowanie algorytmu backtracking do automatycznego znalezienia rozwiązania.

Wyświetlenie Rozwiązania: Prezentacja gotowej i poprawnej planszy Sudoku w czytelnej, sformatowanej formie.

Obsługa Błędów: Zapewnienie komunikatu dla użytkownika, jeśli podana plansza okaże się nierozwiążalna.

## **4. Wybór Technologii i Strategia Rozwiązywania**

### **4.1. Środowisko i Technologia**

Język Programowania: C#

Środowisko: Visual Studio

#### *4.2. Wybór Algorytmu: Backtracking*

Do rozwiązania problemu zostanie wykorzystany algorytm backtracking . Jest to metoda przeszukiwania z nawrotami.

Mechanizm Działania Algorytmu:

Znajdź pierwsze puste pole na planszy.

Wypróbuj w nim kolejno cyfry od 1 do 9.

Dla każdej próby sprawdź, czy wstawienie cyfry jest zgodne z regułami Sudoku (walidacja wiersza, kolumny i bloku  $3 \times 3$ ).

Jeśli cyfra jest poprawna, rekurencyjnie przejdź do następnego pustego pola.

Jeśli rekurencyjne wywołanie zwróci sukces (czyli plansza została rozwiązana), zwróć sukces.

Jeśli żadna z cyfr (1-9) nie prowadzi do rozwiązania (tzw. impas), wyzeruj aktualne pole i cofa się (nawróć) do poprzedniej komórki, aby wypróbować inną cyfrę.

#### *4.3. Złożoność Algorytmu*

Teoretyczna złożoność czasowa algorytmu backtracking w najgorszym przypadku wynosi  $O(9^{\text{do potęgi } n})$ , gdzie  $n$  to liczba pustych pól. Rzeczywista wydajność jest znacznie lepsza niż wskazuje na to teoretyczna złożoność.

### **5. Oczekiwane Rezultaty**

Rezultatem projektu będzie działająca funkcja do rozwiązywania Sudoku w C#. Oczekuje się, że program:

- Będzie poprawnie weryfikować dane wejściowe.
- Będzie znajdować rozwiązanie dla wszystkich poprawnych i rozwiązalnych plansz Sudoku.