****

**Eötvös Loránd Tudományegyetem   
Informatikai Kar   
Programozási Nyelvek és Fordítóprogramok Tanszék**

**Sudoku heurisztikák implementációja**

**Témavezető: Szerző:**Pataki Norbert László Gábor  
Docens, PhD Programtervező Informatikus BSc

Budapest, 2022

Tartalomjegyzék

[1. Bevezető 4](#_Toc89862548)

[1.1. Sudoku 4](#_Toc89862549)

[2. Felhasználói Dokumentáció 5](#_Toc89862550)

[2.1. Rendszerkövetelmények 5](#_Toc89862551)

[2.2. A program futtatása 5](#_Toc89862552)

[2.3. A program használata 5](#_Toc89862553)

[2.3.1. Fájl betöltés 5](#_Toc89862554)

[2.3.2. Heurisztikák kiválasztása 5](#_Toc89862555)

[2.3.3. Tábla használata 5](#_Toc89862556)

[2.3.4. Megoldás Lépések kezelése 5](#_Toc89862557)

[2.4. Alkalmazható heurisztikák 5](#_Toc89862558)

[2.4.1. Naked Singles 5](#_Toc89862559)

[2.4.2. Hidden Singles 5](#_Toc89862560)

[2.4.3. Pointing pairs 5](#_Toc89862561)

[2.4.4. Box line reduction 5](#_Toc89862562)

[2.4.5. Naked subsets 5](#_Toc89862563)

[2.4.6. Hidden subsets 5](#_Toc89862564)

[2.4.7. X-wing 5](#_Toc89862565)

[2.4.8. Swordfish 5](#_Toc89862566)

[3. Fejlesztői dokumentáció 6](#_Toc89862567)

[3.1. Megoldási Terv 6](#_Toc89862568)

[3.2. Megvalósítás 6](#_Toc89862569)

[3.2.1. Sudoku\_History 6](#_Toc89862570)

[3.2.2. Sudoku\_Board 6](#_Toc89862571)

[3.2.3. Sudoku\_Solver 6](#_Toc89862572)

[3.2.4. Sudoku\_Modell 6](#_Toc89862573)

[3.2.5. Sudoku\_Window 6](#_Toc89862574)

[3.2.6. Tesztek 6](#_Toc89862575)

[3.3. Tesztelés 6](#_Toc89862576)

[4. Összegzés 7](#_Toc89862577)

# Bevezető

A szakdolgozatom fő célja egy Sudoku rejtvény megoldó program. A Sudoku rejtvényeknek több változata is van, mi a hagyományos 9x9-es táblát fogjuk használni. A programban kiválaszthatjuk, hogy milyen heurisztikákkal próbálja meg megoldani a rejtvényt. Rejtvényeket tölthetünk be fájlból vagy akár kézzel megadva. A megoldás végén meg tudjuk nézni, hogy melyik heurisztikákat milyen sorrendben alkalmazta a program és akár vissza is tudjuk léptetni egy előzőleges állásra.

## Sudoku

A 19. században francia újságokban szerepeltek különböző változatokban és később 1979-től rejtvény könyvekben is megjelent „Number Place” néven. Japánban nagy népszerűségre tett szert 1979-ben, mikor a Nikoli cég Sukoku néven kiadta a japán változatát. Nemzetközi sikerre 2004-ben tett szerint, mikor a The Times brit magazin megjelentette első változatát az újságjaiban.

A hagyományos Sudoku-ban egy 9x9-es táblát kell kitölteni, ami fel van bontva 3x3-as blokkokra. Miden sorban, oszlopban vagy blokkban egy szám csak egyszer szerepelhet. A még ki nem töltött mezőkben jelölni szoktuk a lehetséges számokat, ezeket jelölteknek fogjuk hívni. A megoldhatóság kedvéért előre kiszoktak tölteni néhány szükséges cellát számokkal úgy, hogy lehetőleg csak egy megoldása létezzen a rejtvények.

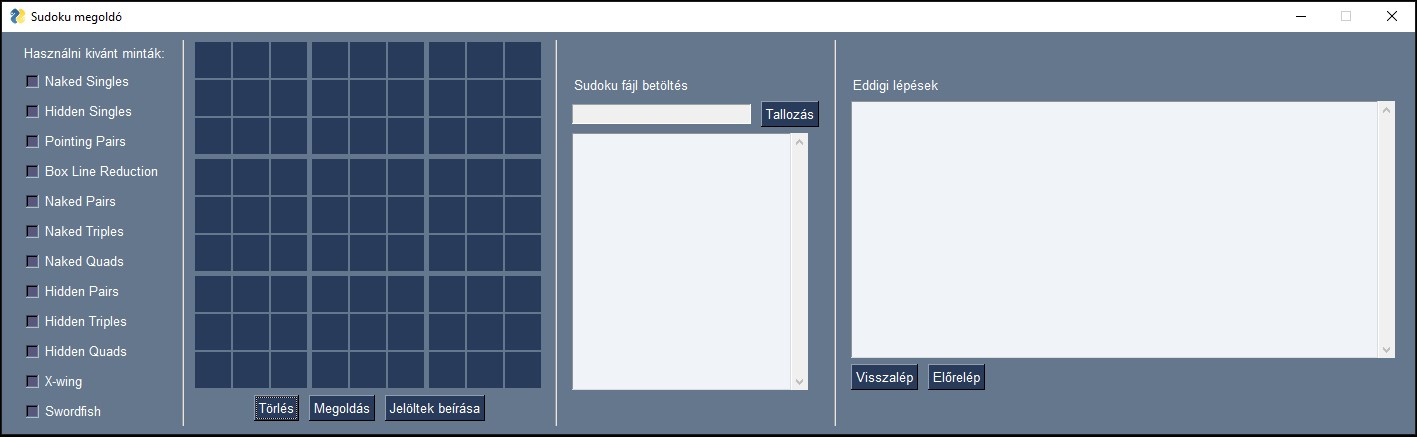
# Felhasználói Dokumentáció

## Rendszerkövetelmények

A programot Windows 10 operációs rendszeren, Python programozási nyelven fejlesztettem. A szoftver futtatásához legalább a Python 3.9-es verziójára van szükség. A számolásokhoz a NumPy és a grafikus felülethez a PySimpleGUI könyvtárakra van szükség.

## A program futtatása

## A program használata

A program futtatása során megjelenő grafikus felület 4 szekcióra van bontva funkcionalitásuk alapján.

2.1. ábra. A program főpanelje

### Fájl betöltés

Sudoku rejtvényeket JSON fájlból tudunk betölteni, aminek a szerkezete a … ábrán látható. A „Tallózás” gombra kattintva a megjelenő ablakban ki tudjuk választani a rejtvényeket tartalmazó JSON fájlunkat, ha rossz kiterjesztésű fájlt vagy rossz szerkezetű JSON fájlt akarunk betölteni akkor a program egy felugró ablakban értesít a fájl helytelenségéről és nem tölt be semmit. A betöltés után megjelenik az alatta lévő dobozban a fájlban tárolt rejtvények neve. A nevek között kattintva tudjuk a táblában váltogatni az egyes rejtvényeket.

### Heurisztikák kiválasztása

2.2. ábra. Példa helyes JSON fájlra

Itt választhatjuk ki a megoldáshoz használandó heurisztikákat, melyek az alábbiak lehetnek:

* naked singles
* hidden singles
* pointing pairs
* box line reduction
* naked subsets (pairs, triples, quads)
* hidden subsets (pairs, triples, quads)
* x-wing
* swordfish

A jelölőnégyzet bejelölésével tudjuk kiválasztani a használandó heurisztikákat. Alaphelyzetben az egyik jelölőnégyzet sincs bejelölve.

### Tábla használata

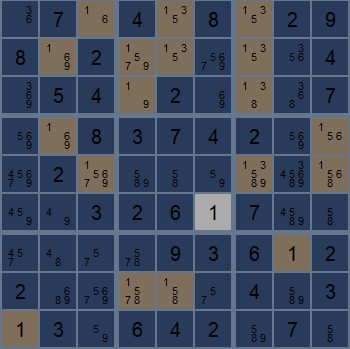
Táblának nevezzük a 9x9 db mezőből álló felületet, ez fogja reprezentálni a Sudoku rejtvényünket. Alatta 3 funkció gomb található „Törlés”, „Megoldás” és „Jelöltek beírása” néven.

Rejtvényt megadni lehet a Sudoku szabályait követve kézzel egy mező kiválasztása után a 1-9-es billentyűk megnyomásával vagy a már fájlból betöltött rejtvények közül választva. Az előre megadott rejtvények közül választva is van lehetőségünk a szabályokat követve módosítani a táblán. Ha egy mező érékét törölni kívánnánk azt a DEL billentyű megnyomásával tudjuk elérni.

A „Törlés” gomb megnyomásával tudjuk a táblát, a jelölőnégyzeteket és a megoldás lépéseit alaphelyzetbe állítani.

A „Megoldás” gomb megnyomásával lezároljuk a tábla összes mezőjét és a program elkezdi alkalmazni a kiválasztott heurisztikákat fentről-lefelé. A listából, ha talál alkalmazható heurisztikát, akkor azt alkalmazza a rejtvényen majd újra elkezd a lista elejéről keresni. A futtatás akkor ér véget, ha minden mezőnek megtalálta a megoldását vagy már nem talált több alkalmazható heurisztikát, ekkor a rejtvényt félkészen hagyja. A futtatás végeztével összes jelölt törlése és megoldás beírása műveletet elmenti az eddigi lépések táblában.

A „Jelöltek beírása” gomb megnyomásával az aktuális tábla alapján az összes jelöltet beírja a helyére, a … ábrán láthatjuk működés közben.

A tábla megoldást tartalmazó mezőire kattintva a program az összes ilyen számot vagy jelöltet tartalmazó mezőt kiemel, amit a … ábrán láthatunk.

2.3. ábra. Sudoku rejtvény kitöltve jelöltekkel és mező kiemeléssel

### Megoldás Lépések kezelése

A program a megoldás alatt az összes használt jelölt törlése vagy megoldás beírása lépést eltárolja. Minden egyes lépés tartalmazza a használt heurisztika nevét, a művelet típusát a módosítás helyét és értékét.

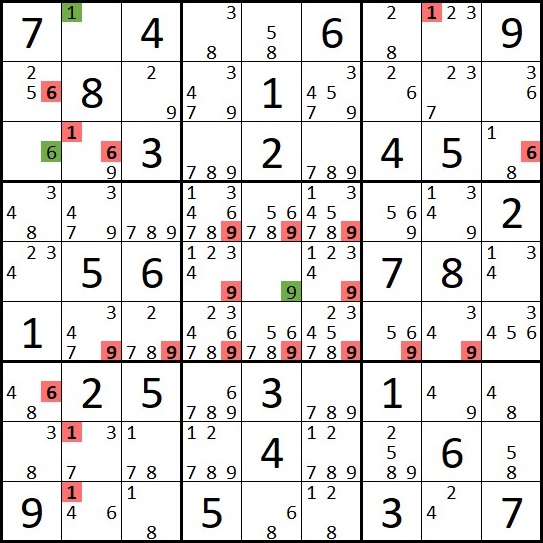
A megoldás befejeztével lehetőségünk van a „Vissza lép” gomb megnyomásával egyesével visszakövetni a megtett lépéseket.

Az „Előre lép” gomb megnyomásával a visszalépés után vissza tudunk lépni a teljes vagy egy részleges megoldásra.

## Alkalmazható heurisztikák

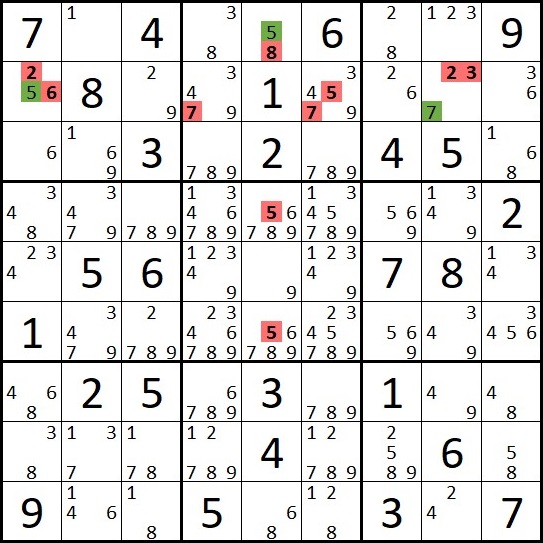
A könnyebb megértéshez a példákban a sakkhoz hasonlóan fogunk a mezőkre hivatkozni, minden sort fentről-lefele egy A-I betű, míg minden oszlopot balról-jobbra egy 1-9 szám fog azonosítani, és így a betű-szám párok fogják jelölni a mezőket. Ezenfelül minden jelöltet fel is fogunk tüntetni.

### Naked Singles

Olyan mezőket keresünk, ahol csak egyetlen egy jelölt található, ekkor a mezőbe beírjuk a jelöltet, mint megoldást és a mezővel megegyező sorból, oszlopból és blokkból eltávolítjuk ezt a számot a jelöltek közül.

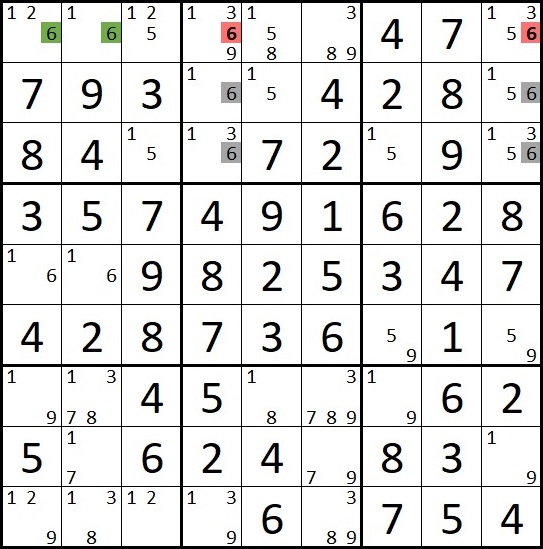
2.4. ábra Naked Singles

### Hidden Singles

Olyan jelölteket keresünk, ami egy sorban, oszlopban vagy blokkban csak egyszer szerepel, ebben a mezőben több jelölt is lehet, ekkor ezt a jelöltet beírjuk a mezőjébe, mint megoldást és a vele megegyező sorból, oszlopból és blokkból eltávolítjuk ezt a számot a jelöltek közül.

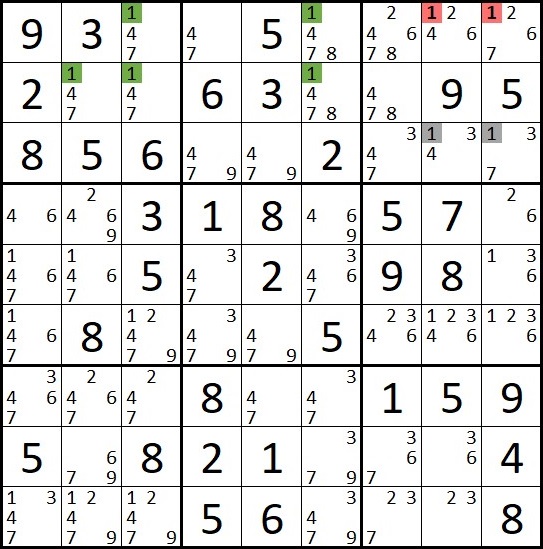
2.5. ábra. Hidden singles

### Pointing pairs

Olyan számokat keresünk, ami egy blokkon belül csak egy sorban vagy oszlopban szerepel, mint jelölt és ekkor a blokkon kívül az adott sorból vagy oszlopból töröljük ezt a számot a jelöltek közül.

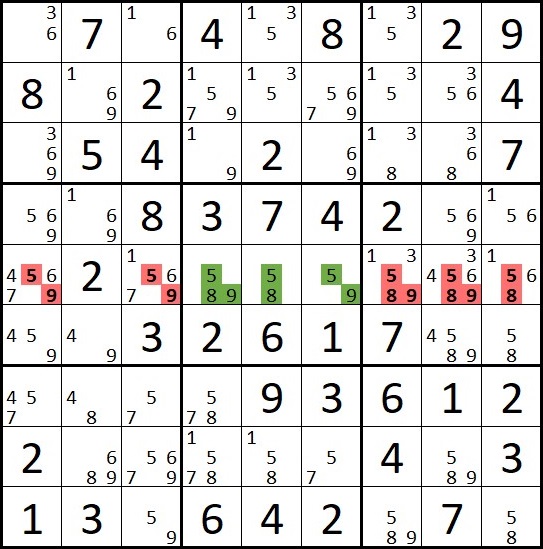
2.6. ábra. Pointing Pairs

### Box line reduction

Olyan sorokat vagy oszlopokat keresünk, ahol egy szám a jelöltek között csak egy blokkban szerepel, ekkor ezt a számot törölhetjük a blokk jelöltjei közül meghagyva a kereset sorban vagy oszlopban.

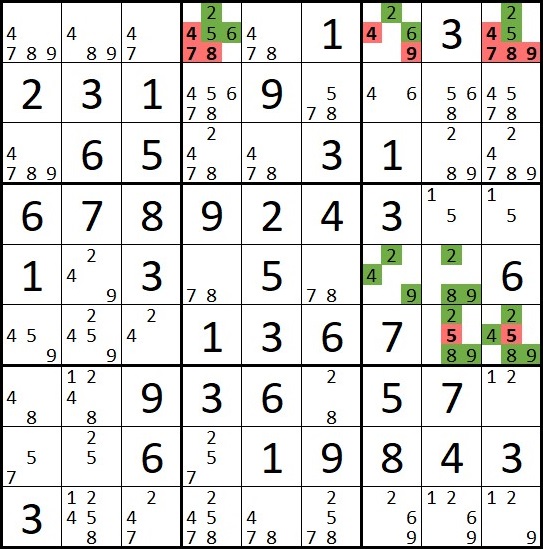
2.7. ábra. Box Line Reduction

### Naked subsets

Olyan 2,3 vagy 4 db számot keresünk a jelöltek között, hogy egy sor, oszlop vagy blokknak a 2,3 vagy 4 mezőjében csak ezek a számok szerepeljenek, ekkor a sor, oszlop vagy blokk minden más mezőjéből ezeket a számokat töröljük a jelöltek közül.

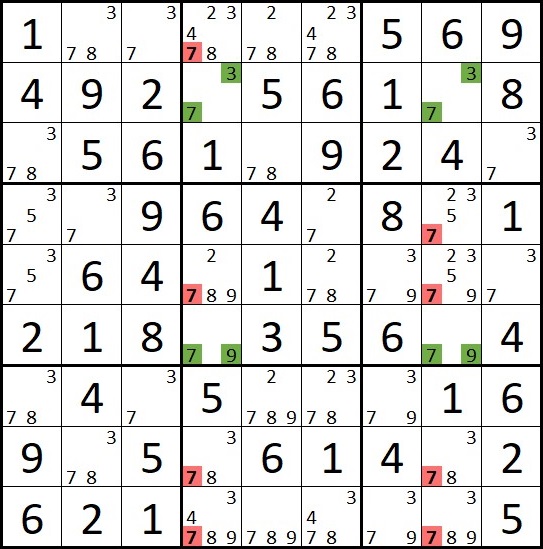
2.8. ábra. Naked Triples

### Hidden subsets

Olyan 2,3 vagy 4 db számot keresünk a jelöltek között, hogy az egy sor, oszlop vagy blokknak csak 2,3 vagy 4 mezőjében szerepeljenek, más számok is szerepelhetnek ezekben a mezőkben, ekkor minden más jelöltet törlünk ezekből a mezőkből.

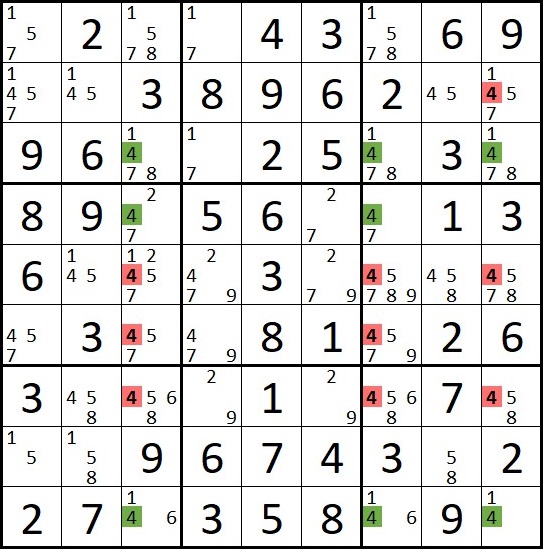
2.9. ábra. Hidden Triples and Quads

### X-wing

Olyan 2 sort vagy oszlopot keresünk, ahol egy szám csak 2 mezőben található meg és ezek a mezők ugyanabban az oszlopba vagy sorba esnek, ekkor az oszlop vagy sor minden másik mezőjéből törölhetjük a jelöltek közül a számot.

2.10. ábra. X-wing

### Swordfish

Olyan 3 sort vagy oszlopot keresünk, ahol egy szám csak 2 vagy 3 mezőben található meg és ezek a mezők ugyanabban az oszlopba vagy sorba esnek, ekkor az oszlop vagy sor minden másik mezőjéből törölhetjük a jelöltek közül a számot.

2.11. ábra. Swordfish

# Fejlesztői dokumentáció

## Megoldási Terv

## Megvalósítás

### Sudoku\_History

### Sudoku\_Board

### Sudoku\_Solver

### Sudoku\_Modell

### Sudoku\_Window

### Tesztek

## Tesztelés

# Összegzés