

# Sudoku Solver

Sudoku je logická hra, ktorá sa skladá z mriežky 9x9 s číslicami od 1 po 9. Cieľom hry je doplniť chýbajúce číslice podľa troch základných pravidiel:

1. každý riadok môže obsahovať každú číslicu iba raz
2. každý stĺpec môže obsahovať každú číslicu iba raz
3. mriežka je rozdelená do podoblastí 3x3, ktoré môžu obsahovať každú číslicu iba raz.

Na začiatku mriežka obsahuje číslice tak, aby pre mriežku existovalo iba jedno riešenie. Ľudia riešia sudoku hry pomocou logických indukcií, napr. pre:

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

číslica 2 sa môže v pravej strednej podoblasti nachádzať iba v jednom poli:

|   |   |   |   |   |   |              |   |   |
|---|---|---|---|---|---|--------------|---|---|
|   |   |   |   |   |   | <del>8</del> |   | 2 |
|   | 9 | 8 | 4 |   | 2 |              |   |   |
| 2 |   | 5 |   |   |   |              | 6 |   |
|   | 4 | 1 |   | 8 |   |              |   | 7 |
|   | 5 |   | 6 | 2 | 4 |              | 3 |   |
| 6 |   |   |   | 3 |   | <del>9</del> | 4 |   |
|   | 7 |   |   |   |   | 2            |   | 6 |
|   |   |   | 8 |   | 3 | <del>5</del> | 1 |   |
| 8 |   | 9 |   |   |   |              |   |   |

Takéto implikácie sa dajú ale iba ťažko naprogramovať vzhľadom na to, že niekedy vyžadujú uvažovať niekoľko krokov dopredu. Pre počítače je viac vhodné riešenie pomocou brute force algoritmu, v ktorom počítač rekurzívne vyskúša všetky možnosti až kým nenájde riešenie alebo

nenarazí na porušenie niektorého pravidla, v tomto prípade vráti všetky zmeny a začne ďalšie kolo rekurzívnych testov. Napríklad, v nižšie uvedenom príklade zo strednej hornej oblasti chýbajú dve čísla, 1 a 7. Algoritmus môže do modrého štvorca zapísať číslu 1, čo je validná možnosť, ale v ďalšom kroku spôsobí porušenie pravidla (dve 7 v jednom stĺpci). Preto algoritmus musí vymazať 1 a skúšať druhú možnosť, t. j. 7 a takto nájde správne riešenie pre podoblasť.

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

### Úloha 1 – 1 bod

Implementujte funkciu `loadSudoku`, ktorá má jediný parameter, reprezentáciu mriežky pomocou stringu, kde prázdne bunky sú reprezentované 0. Napríklad vyššie uvedený príklad bez hráčom vyplnených čísiel môže byť reprezentovaný nasledovne:

```
00194000800000200000008500600750800000000600900497103800603009000
02504800807000503
```

Ďalšie príklady pre testovanie nájdete v súbore `sudokus.txt`.

Funkcia `loadSudoku` vracia jednu hodnotu, mriežku reprezentovanú ako zoznam zoznamov.

**Pozor!** Pri hodnotení vášho riešenia sa bude kontrolovať aj schopnosť funkcie `loadSudoku` spracovať nesprávny vstup – t. j. funkcia by mala upozorniť používateľa na nesprávne definovanú úlohu sudoku.

### Úloha 2 – 0,5 boda

Implementujte funkciu `findNextEmptyCell`, ktorá nájde prvú nevyplnenú bunku v mriežke a vráti jej súradnice. Napríklad, pre vyššie uvedený príklad by vrátila hodnoty 0, 0 keďže prvá bunka nie je vyplnená. Ak mriežka už neobsahuje prázdnu bunku, funkcia vráti hodnoty -1, -1. Funkcia má jeden parameter – mriežku reprezentovanú ako zoznam zoznamov.

### Úloha 3 – 0,5 boda

Implementujte funkciu `isSolved`, ktorá vráti `True` ak mriežka je (správne) vyriešená a `False` naopak. Funkcia má jeden parameter – mriežku reprezentovanú ako zoznam zoznamov.

#### Úloha 4 – 0,5 boda

Implementujte funkciu `isValid`, ktorá má štyri parametre: `grid` (reprezentuje mriežku), `i`, `j`, a `e`. Funkcia vracia `True` ak do mriežky `grid` je možné na pozíciu `[i, j]` zapísať číslu `e` a `False` naopak. Hodnota `e` musí byť číslu od 1 po 9, `i` a `j` musia byť z intervalu 0 až 8.

#### Úloha 5 – 0,5 boda

Implementujte funkciu `printSudoku`, ktorá pomocou príkazu `print` vypíše do konzoly mriežku sudoku. Funkcia má jeden parameter – mriežku reprezentovanú ako zoznam zoznamov.

#### Úloha 6 – 2 bod

Implementujte funkciu `solveSudoku`, ktorá má jeden parameter – stringovú reprezentáciu úlohy sudoku. Funkcia získa reprezentáciu mriežky pomocou zoznamov (použite funkciu `loadSudoku`) a následne úlohu vyrieši. Pri riešení použite rekurzívne vyskúšanie všetkých možností ako bolo uvedené vyššie.

Funkcia vráti vyriešenú úlohu (teda mriežku bez prázdnych buniek).

#### Úloha 7 – 1 bod

Implementujte funkciu `fillSure`, ktorá má jeden parameter – reprezentáciu mriežky pomocou zoznamov. Funkcia napodobňuje spôsob ktorým ľudia riešia úlohy sudoku, t. j. hľadá bunky do ktorých je možné zapísať iba jednu číslu, alebo také riadky, stĺpce alebo podoblasti, v ktorých niektorú číslu je možné zapísať iba do jednej bunky. Funkcia vráti mriežku s doplnenými hodnotami a slúži pre zjednodušenie úlohy pred rekurzívnym vyhľadávaním, takže zavolajte ju vo funkcii `solveSudoku` po načítaní mriežky.

Je na vás, aké implikácie funkcia obsahuje, musí ale obsahovať aspoň jednu implikáciu. Pre ďalšie implementované implikácie môžete získať bonusové body (závisí od ich zložitosti).