

# SuSol

**Programátorská dokumentace  
solveru a generátoru**

**Solver a generátor použitý v projektu je zcela jistě použitelný pro další vývojáře v Pythonu. Proto jsou všechny funkce zdokumentovány tak, aby každý věděl, jak jich využít.**

## **OBSAH:**

|                       |          |
|-----------------------|----------|
| <b>Solver.....</b>    | <b>3</b> |
| <b>Generátor.....</b> | <b>9</b> |

# SOLVER

Pro použití solveru je potřeba nainportovat `solver` resp. `solver2`. Soubory `solver.py` a `solver2.py` je třeba mít buď ve stejné složce jako projekt, nebo v podsložce Pythonu `site-packages`.

## solver.py

Balíček obsahující funkce k řešení sudoku. Tento solver je zastaralý. V novější verzi je naimplementováno vše přehledněji a rychleji, je doporučeno proto používat `solver2.py`. Tento solver dokáže efektivně nalézt náhodná řešení (což ten nový neumí).

**`solver.solve(raw, mode=0, bf=True)`**

Vyřeší sudoku.

`int[][] raw` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

`int mode` celé číslo znázorňující počet řešení, které má solver najít. Pokud je rovno 0, solver nalezne všechna řešení. Pokud je záporné, solver nepostupuje systematicky a vrátí daný počet náhodných řešení (například -3 vrátí tři náhodná řešení, -1 a prázdná sudoku mřížka vrátí jedno náhodné z mnoha možných vyplnění mřížky).

`boolean bf` pokud je `False`, solver bude řešit sudoku pouze strategií Naked Single (Hidden Single není v této verzi solveru implementována). Pokud se sudoku takto nedá dořešit až do konce, sudoku zůstane nedořešeno. Pokud je `True`, toto nedořešené sudoku bude dořešeno hrubou silou.

Funkce vrací pole matic o rozměrech 9x9, přičemž každý prvek tohoto pole představuje jedno řešení a každá matice představuje sudoku. (`int[][][]`)

## solver2.py

Balíček obsahující funkce k řešení sudoku. Tento solver je aktuální, pokud nejsou potřeba náhodná řešení, je doporučeno používat tento.

## **`solver2.zkontrolovatVstup(sudoku)`**

Zkontroluje, zda je sudoku platné.

`sudoku` Vstupní parametr funkce.

Funkce vrací `True`, pokud je vstupní parametr maticí o rozměrech 9x9, obsahující celá čísla od 0 do 9. Jinak vrací `False`. (**`boolean`**)

## **`solver2.inicializovatKandidaty(sudoku)`**

Vygeneruje pro každé nezadané políčko seznam kandidátů od 1 do 9 (nepročištěný).

`int[][] sudoku` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

Funkce vrací matici 9x9 devítiprvkových polí `[1, 2, 3, 4, 5, 6, 7, 8, 9]` pro každé políčko, které ve vstupním sudoku není zadáno. (**`int[][][]`**)

## **`solver2.generujKandidaty(kandidati, reseni)`**

Vymaže z daného seznamu kandidátů taková čísla, která jsou v daném sudoku zadána v rámci některého ze setů, ve kterém se právě zkoumané políčko nachází.

`int[][][] kandidati` matice o rozměrech 9x9, obsahující pole celých čísel, představující seznam kandidátů pro dané políčko.

`int[][] reseni` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

Funkce vrací matici 9x9 polí celých čísel, přičemž každé pole představuje reálný seznam kandidátů pro dané políčko – jsou ořezána všechna čísla, které se nachází ve stejném řádku, sloupci nebo čtverci. (**`int[][][]`**)

### **`solver2.nakedSingle(kandidati, reseni, logovatPostup=False)`**

Ze zadaných kandidátů a sudoku najde aplikaci strategie Naked Single, provede ji a vrátí sudoku a seznam kandidátů po této operaci, volitelně i popis tohoto kroku. Zároveň funkce v určitých případech umí odhalit sudoku, které nemá řešení. Tento jev také oznámí.

`int[][][] kandidati` matice o rozměrech 9x9, obsahující pole celých čísel, představující seznam kandidátů pro dané políčko.

`int[][] reseni` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

`boolean logovatPostup` Pokud je `False`, funkce nevrátí popis kroku (poslední prvek vráceného pole)

Funkce vrací tří/čtyřprvkové pole: prvním prvkem je aktualizovaný seznam kandidátů po operaci, druhým prvkem aktualizované řešení po operaci, třetím prvkem je `boolean: True` pokud sudoku má řešení, `False` pokud nemá; volitelně čtvrtým prvkem je popis kroku s následující syntaxí:

```
[str "Naked Single", [int sloupec, int radek], int doplnene_cislo]
```

slovy jméno strategie, souřadnice políčka a doplňované číslo. ([1])

### **`solver2.hiddenSingle(kandidati, reseni, logovatPostup=False)`**

Ze zadaných kandidátů a sudoku najde aplikaci strategie Hidden Single, provede ji a vrátí sudoku a seznam kandidátů po této operaci, volitelně i popis tohoto kroku. Zároveň funkce v určitých případech umí odhalit sudoku, které nemá řešení. Tento jev také oznámí.

`int[][][] kandidati` matice o rozměrech 9x9, obsahující pole celých čísel, představující seznam kandidátů pro dané políčko.

`int[][] reseni` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

`boolean logovatPostup` Pokud je `False`, funkce nevrátí popis kroku (poslední prvek vráceného pole)

Funkce vrací tří/čtyřprvkové pole: prvním prvkem je aktualizovaný seznam kandidátů po operaci, druhým prvkem aktualizované řešení po operaci, třetím prvkem je `boolean: True` pokud sudoku má řešení, `False` pokud nemá; volitelně čtvrtým prvkem je popis kroku s následující syntaxí:

```
[str "Hidden Single",[str typ_setu,int poradi_setu],[int sloupec,int radek],int doplnene_cislo]
```

slovy jméno strategie, řádek/sloupec/čtverec ("`r`", "`s`" nebo "`c`") a jeho pořadové číslo, souřadnice políčka a doplňované číslo. (`[]`)

### **`solver2.sudokuVyreseno(reseni)`**

Zjistí, zda je dané sudoku vyřešeno (neobsahuje nevyplněná políčka), ale nekontroluje správnost řešení. Pro zjištění správnosti je třeba volat `solver2.solvePC` s parametrem hotového sudoku.

`int[][] reseni` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

Funkce vrací `True`, pokud sudoku neobsahuje nevyplněná políčka. V opačném případě vrací `False`. (**`boolean`**)

### **`solver2.najdiPolickoProTest(kandidati)`**

Najde nevyplněné políčko s nejméně kandidáty a vrátí jeho souřadnice. Pokud je takovýchto políček více, vrátí to, které je při procházení po řádcích z levého horního rohu tomuto místu nejbližší. Pokud jsou již všechna políčka vyplněna, vrátí `[-1, -1]`.

`int[][][] kandidati` matice o rozměrech 9x9, obsahující pole celých čísel, představující seznam kandidátů pro dané políčko.

Funkce vrací dvouprvkové pole celých čísel představující souřadnice políčka. `[-1, -1]` znamená, že odpovídající políčko nebylo nalezeno. (**`int[][]`**)

## **`solver2.bruteForce(reseni,pocetReseni)`**

Vyřeší sudoku hrubou silou.

`int[][] reseni` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

`int pocetReseni` celé číslo. Pokud je větší nebo rovno 1, znamená počet řešení, které se bude solver snažit najít. Pokud jich má dané sudoku méně než tento počet, budou nalezena všechna řešení. Pokud je tento argument záporný nebo roven nule, budou nalezena všechna řešení.

Funkce vrací pole matic o rozměrech 9x9, přičemž každý prvek tohoto pole představuje jedno řešení a každá matice představuje sudoku. (`int[][][]`)

## **`solver2.solvePC(zad,pocetReseni=1,bf=True)`**

Vyřeší sudoku.

`int[][] zad` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

`int pocetReseni` celé číslo. Pokud je větší nebo rovno 1, znamená počet řešení, které se bude solver snažit najít. Pokud jich má dané sudoku méně než tento počet, budou nalezena všechna řešení. Pokud je tento argument záporný nebo roven nule, budou nalezena všechna řešení.

`boolean bf` pokud je `False`, sudoku se bude řešit pouze strategiemi Naked Single a Hidden Single. Pokud se sudoku takto nedá dořešit až do konce, sudoku zůstane nedořešeno. Pokud je `True`, toto nedořešené sudoku bude dořešeno hrubou silou.

Funkce vrací dvouprvkové pole: prvním prvkem je pole matic o rozměrech 9x9, přičemž každý prvek tohoto pole představuje jedno řešení a každá matice představuje sudoku. Druhým prvkem je `float` představující čas v milisekundách potřebný pro provedení procedury. (`[]`)

**`solver2.solveHuman(zad,kandidati=None,  
naked_single=True,hidden_single=True)`**

Nalezne krok při řešení sudoku pomocí strategií Naked Single a Hidden Single.

`int[][] zad` matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko.

`int[][][] kandidati` matice o rozměrech 9x9, obsahující pole celých čísel, představující seznam kandidátů pro dané políčko. Pokud není specifikováno, funkce si jej vygeneruje sama.

`boolean naked_single` pokud je `False`, strategie Naked Single nebude aplikována. Pokud je `True`, strategie Naked Single bude aplikována. Pozor! Nastavení tohoto parametru na `False` může způsobit dlouhou výpočetní dobu, ve výjimečných případech dokonce zacyklení.

`boolean hidden_single` pokud je `False`, strategie Hidden Single nebude aplikována. Pokud je `True`, strategie Hidden Single bude aplikována. Pozor! Nastavení tohoto parametru na `False` může způsobit dlouhou výpočetní dobu, ve výjimečných případech dokonce zacyklení.

Funkce vrací čtyřprvkové pole: prvním prvkem je pole popisu kroku (specifikováno níže), druhým prvkem je `float` čas v milisekundách potřebný k řešení, třetím prvkem je `boolean` znázorňující, zda sudoku má řešení (`True` – má, `False` – nemá) a čtvrtým prvkem je `boolean` znázorňující, zda je sudoku již vyřešeno (`True` – je vyřešeno, `False` – není vyřešeno).

Popis kroku Naked Single:

`[str "Naked Single",[int sloupec,int radek],int doplnene_cislo]`  
slovy jméno strategie, souřadnice políčka a doplňované číslo.

Popis kroku Hidden Single:

`[str "Hidden Single",[str typ_setu,int poradi_setu],[int sloupec,int radek],int doplnene_cislo]`  
slovy jméno strategie, řádek/sloupec/čtverec ("r","s" nebo "c") a jeho pořadové číslo, souřadnice políčka a doplňované číslo. ([1])



# GENERÁTOR

Pro použití solveru je potřeba nainportovat `generator`. Soubor `generator.py` je třeba mít buď ve stejné složce jako projekt, nebo v podsložce Pythonu `site-packages`.

## generator.py

Balíček obsahující funkce ke generování sudoku.

### **generator.vyrobItMrizku()**

Vyrobí podle pravidel sudoku náhodně vyplněnou mřížku.

Funkce vrací matici o rozměrech 9x9 celých čísel od 1 do 9 představující sudoku.

(int[][])

### **generator.generate(raw=None, singlesol=True, limit=0, bf=True)**

Vygeneruje sudoku připravené k řešení.

int[][] raw matice o rozměrech 9x9, obsahující celá čísla od 0 do 9. Představuje zadané sudoku s tím, že 0 představuje prázdné políčko. Bude použita jako základ pro generování, může mít prázdná místa nebo ne. Pokud bude předáno takové sudoku, které samo o sobě není jednoznačné, generátor vrátí totožnou matici. Pokud není specifikována vůbec, funkce si matici sama náhodně vytvoří.

boolean singlesol pokud je True, generátor garantuje, že zadané sudoku bude mít právě jedno řešení. Výjimkou je, když se generátoru jako základ předá nejednoznačné sudoku.

int limit celé číslo značící, kolik minimálně čísel má ve vygenerovaném sudoku zůstat zadaných. Pokud je větší nebo rovno 81 nebo záporné, bude limit ignorován.

boolean bf pokud je False, při testu jednoznačnosti řešení nebude použito řešení sudoku hrubou silou, ale pouze pomocí Naked Single a Hidden Single. Výsledné sudoku bude proto trochu jednodušší. Pokud je True, sudoku bude řešeno hrubou silou, a proto může být pro luštitelé obtížnější.

Funkce vrací matici o rozměrech 9x9 celých čísel od 0 do 9 představující sudoku, přičemž 0 znamená prázdné políčko. (int[][])