

# Τρίτη εργασία στις “Αρχές Γλωσσών Προγραμματισμού”

## Δηλωτικός Προγραμματισμός

Γκόγκος Χρήστος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

Άρτα, Μάιος 2021

### Εισαγωγή

Ο στόχος της εργασίας είναι να αποκτηθεί εξοικείωση με το δηλωτικό προγραμματισμό και ιδιαίτερα με την επίλυση προβλημάτων στα οποία ζητείται ο εντοπισμός λύσεων μέσα σε έναν πιθανά τεράστιο αλλά απαριθμήσιμο σύνολο πιθανών λύσεων. Για την επίλυση της εργασίας θα χρησιμοποιηθεί το λογισμικό Google OR-Tools και ειδικά ο επιλυτής του CP-SAT.

### Περιγραφή προβλήματος

#### Sudoku

Το Sudoku είναι ένα πολύ δημοφιλές παιχνίδι στο οποίο δίνεται ένα ταμπλό  $9 \times 9$  που περιέχει σε κάποιες από τις θέσεις του συμπληρωμένα ψηφία από το 1 μέχρι και το 9. Στόχος είναι να συμπληρωθεί πλήρως το ταμπλό (επίσης επιλέγοντας ψηφία από το 1 μέχρι και το 9) έτσι ώστε σε κάθε γραμμή, σε κάθε στήλη και σε καθένα από τα 9 υποταμπλό  $3 \times 3$  που σχηματίζονται από πάνω αριστερά μέχρι και κάτω δεξιά του αρχικού ταμπλό να περιέχονται όλα τα ψηφία από το 1 μέχρι και το 9. Στο Σχήμα 1 παρουσιάζεται ένα Sudoku και η λύση του.

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

Σχήμα 1: Ένα ταμπλό Sudoku και η λύση του (από το How Sudoku could win you a million dollars)

### Ζητούμενα

#### Ανάγνωση προβλημάτων

Ζητείται η συγγραφή ενός προγράμματος σε Python που να μπορεί να διαβάσει προβλήματα Sudoku. Κάθε πρόβλημα θα δίνεται ως ένα λεκτικό 81 ψηφίων που θα περιέχει για τις θέσεις που περιέχουν ψηφίο, το ίδιο το ψηφίο και για τις θέσεις που δεν είναι συμπληρωμένες το ψηφίο μηδέν. Η σειρά παράθεσης των ψηφίων θα είναι από από αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Για παράδειγμα, το λεκτικό εισόδου για το Σχήμα 2 θα είναι το 310004069000000200008005040000000005006000017807030000590700006600003050000100002.

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

Σχήμα 2: Ένα ταμπλό Sudoku στην αρχική του μορφή [http://sudopedia.enjoysudoku.com/Diagrams\\_and\\_Notations.html](http://sudopedia.enjoysudoku.com/Diagrams_and_Notations.html)

Το πρόγραμμα εφόσον διαβάσει ένα πρόβλημα θα πρέπει να είναι σε θέση να εμφανίσει το ταμπλό όπως στη συνέχεια.

Listing 1: Απεικόνιση του ταμπλό ως κείμενο

```

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

```

### Εμφάνιση “Pencil Marks”

Ζητείται η εμφάνιση των πιθανών τιμών που μπορούν να τοποθετηθούν σε κάθε θέση εξετάζοντας “απλά” τους 3 περιορισμούς, δηλαδή ότι θα πρέπει σε κάθε γραμμή, στήλη και μπλοκ  $3 \times 3$  να υπάρχουν όλα τα ψηφία από το 1 μέχρι και το 9. Για παράδειγμα για το ταμπλό του σχήματος 2 το ταμπλό που ζητείται είναι το ταμπλό του σχήματος 3.

3	1	2 <sup>5</sup>	2 <sup>2</sup>	2 <sup>2</sup>	4	5 <sup>7 8</sup>	6	9
		4 5 6	4 5	8 <sup>7 8</sup>	3 <sup>1</sup>	1 <sup>6</sup>	3 <sup>1 3</sup>	
		2	2 <sup>6</sup>	8	2 <sup>3</sup>	5 <sup>1 3</sup>	4	1 <sup>3</sup>
1 2 <sup>4</sup>	2 3 <sup>4</sup>	1 2 3 <sup>4</sup>	2 <sup>4</sup>	1 2 <sup>4</sup>	1 2 <sup>4</sup>	3 <sup>4</sup>	2 3 <sup>4</sup>	5
2 <sup>4</sup>	2 3 <sup>4</sup>	6	2 <sup>4</sup>	2 <sup>4</sup>	2 <sup>4</sup>	3 <sup>4</sup>	1	7
8	4 5	7	4 5 6 <sup>9</sup>	3	1 2 <sup>6 9</sup>	4 <sup>6 9</sup>	2 <sup>4</sup>	
5	9	1 2 3 <sup>4</sup>	7	2 <sup>4</sup>	2 <sup>4</sup>	1 3 <sup>4</sup>	3 <sup>4</sup>	6
6	4 7 8	2 <sup>4</sup>	1 2 <sup>4</sup>	2 <sup>4</sup>	2 <sup>4</sup>	3 <sup>4</sup>	5 <sup>1 4</sup>	
		3 <sup>4</sup>	3 <sup>4</sup>	1	4 5 6 <sup>8 9</sup>	6 <sup>4</sup>	3 <sup>4</sup>	2

Σχήμα 3: Το ταμπλό Sudoku με τις πιθανές τιμές σε κάθε θέση [http://sudopedia.enjoysudoku.com/Diagrams\\_and\\_Notations.html](http://sudopedia.enjoysudoku.com/Diagrams_and_Notations.html)

Το πρόγραμμα θα πρέπει να είναι σε θέση να εμφανίσει το ταμπλό με τις πιθανές τιμές για κάθε θέση όπως στη συνέχεια.

Listing 2: Το ταμπλό με pencil-marks

3	1	25	28	278	4	578	6	9
479	4567	459	3689	16789	16789	2	378	138
279	267	8	2369	12679	5	137	4	13
1249	234	12349	24689	1246789	126789	34689	2389	5
249	2345	6	24589	24589	289	3489	1	7
8	245	7	24569	3	1269	469	29	4
5	9	1234	7	248	28	1348	38	6
6	2478	124	2489	2489	3	14789	5	148
47	3478	34	1	45689	689	34789	3789	2

## Επίλυση προβλημάτων με το OR-Tools

Ζητείται η επίλυση του ταμπλό Sudoku που θα έχει φορτωθεί με τη χρήση του CP-SAT επιλυτή του Google OR-Tools. Επιλύστε τα ακόλουθα 20 προβλήματα Sudoku.

Listing 3: 20 προβλήματα Sudoku

```
070000043040009610800634900094052000358460020000800530080070091902100005007040802
301086504046521070500000001400800002080347900009050038004090200008734090007208103
048301560360008090910670003020000935509010200670020010004002107090100008150834029
008317000004205109000040070327160904901450000045700800030001060872604000416070080
040890630000136820800740519000467052450020700267010000520003400010280970004050063
561092730020780090900005046600000427010070003073000819035900670700103080000000050
310450900072986143906010508639178020150090806004003700005731009701829350000645010
800134902041096080005070010008605000406310009023040860500709000010080040000401006
165293004000001632023060090009175000500900018002030049098000006000000950000429381
000003610000015007000008090086000700030800100500120309005060904060900530403701008
405001068073628500009003070240790030006102005950000021507064213080217050612300007
960405100020060504001703006100004000490130050002007601209006038070218905600079000
90452007000189024000264300007096038000010870060000001009008000000750030000312569
001408006093520741000010520602080300007060000005039060064052109020000654500607083
007300054245080900003040070070960000000020760000801002008294016609108020000007003
005346170000000050000800009502930741070000003000700020090050632207600400600420007
320900400705021800001060372218037009500480700000005000670000280000873900804000107
00003000748096050106357082000961020335009700600000509400000005804706910001040070
087002010204017003006800705508001000640008100002050670439180007020900030700023091
040000008760020349000470500900000030000036702308947000000004010200700603690001000
```

## Παράδοση εργασίας

Η εργασία μπορεί να γίνει σε ομάδες των 3 ατόμων και θα πρέπει να παραδοθεί το αργότερο μέχρι τις 11/6/2021 μέσω του ecourse. Στο ecourse θα ανέβει μόνο το URL του GitHub αποθετηρίου και το commit hash code που θα έχει δημιουργηθεί πριν την ημερομηνία παράδοσης της εργασίας.

Η εργασία θα “ανέβει” σε ένα ιδιωτικό αποθετήριο στο GitHub με όνομα DITUOI\_AGP\_SUDOKU\_XXXX όπου XXXX θα είναι ο αριθμός μητρώου του εκπροσώπου της ομάδας, και θα πρέπει να περιέχει:

1. Αρχείο README.md με τα ονόματα των μελών της ομάδας, οδηγίες για την εκτέλεση του κώδικα και παρουσίαση των αποτελεσμάτων των ερωτημάτων.
2. Κώδικα σε Python που υλοποιεί τις απαντήσεις στα ερωτήματα.

Στο ιδιωτικό αποθετήριο που θα δημιουργήσετε, προσθέστε ως συνεργάτες (collaborators) τους chgogos, vasnastos και pint00082.

## Σχετικές ιστοσελίδες

- [https://hackmd.io/@cgogos/Byie\\_58\\_d](https://hackmd.io/@cgogos/Byie_58_d)
- <https://developers.google.com/optimization>
- <https://www.kaggle.com/pintowar/modeling-a-sudoku-solver-with-or-tools>
- [http://sudopedia.enjoysudoku.com/Diagrams\\_and\\_Notations.html](http://sudopedia.enjoysudoku.com/Diagrams_and_Notations.html)