

**ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО**  
**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1**

**2023-2024**

**- други домаћи задатак -**

**Опште напомене:**

1. Домаћи задатак 2 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику *C* или *Python*.
2. Пре одбране, сви студенти раде тест знања који се ради на рачунару коришћењем система *Moodle* (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да креирају налог и пријаве се на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико се студент региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Унос података треба омогућити путем читања из датотеке и/или стандардног улаза.
5. Решења треба да буду отпорна на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
6. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као успешно решење проблема које може освојити максималан број поена.**
7. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити која претпоставка је усвојена (или које претпоставке су усвојене) и која су ограничења програма (на пример, максимална димензија матрице и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Одбрана другог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
9. За решавање задатака који имају више комбинација користити следеће формуле.  
(**R** – редни број индекса, **G** – последње две цифре године уписа):  
$$i = (R + G) \bmod 2 + 1$$
10. Име датотеке која се предаје мора бити **dz2.(c|py)**
11. Предаја домаћих ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене.
12. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака, као и да пријаве теже случајеве повреде Правилника о дисциплинској одговорности студената Универзитета у Београду Дисциплинској комисији Факултета.

## Задатак – Игра „Судоку“ [100 поена]

Игра „Судоку“ представља логичку загонетку постављања бројева, обично представљену матрицом  $9 \times 9$ , тако да се ни у једној врсти, ни у једној колони и ни у једном блоку (обично  $3 \times 3$ ) не понављају исти бројеви, односно свака врста, колона и блок садрже све различите бројеве од 1 до 9. Почетно стање загонетке обично има постављене бројеве у одређеним пољима, а циљ је попунити остатак матрице, тако да горенаведени услови буду испуњени. Више информација о оригиналној игри се може видети на <https://en.wikipedia.org/wiki/Sudoku>.

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

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

Слика 1. Приказ једног почетног стања и решене „Судоку“ загонетке за величину  $9 \times 9$

У циљу решавања овог домаћег задатка, посматраћемо једноставнију варијанту ове игре, а то је варијанта матрице  $4 \times 4$ .

		3	
		1	
			1
3		2	

Слика 2. Приказ једног почетног стања за „Судоку“ загонетке за величине  $4 \times 4$

У варијанти игре  $4 \times 4$ , матрица је подељена на блокове величине  $2 \times 2$  и смештају се бројеви од 1 до 4. У зависности од почетног стања загонетке, игра може имати тачно једно решење, више тачних решења или ниједно решење (нпр. може почетно стање бити невалидно). Такође, у циљу једноставнијег решавања игре, сматрати да свако почетно стање мора имати барем 30% иницијално попуњених поља. У том смислу, сматрати да у игри  $4 \times 4$  најмање 5 поља мора бити попуњено.

**Потребно је имплементирати решавање „Судоку“ загонетки применом стабала одлучивања.**

**[10 поена] Учитавање игре**

Једна игра се започиње читавањем са стандардног улаза или читавањем одговарајућег текстуалног фајла следећег формата:

- У првом реду се задаје димензија матрице  $N$  (овде 4).
- У наредних  $N$  редова се налази почетно стање сваког реда матрице, тако што су празна места у матрици означена бројем 0.

**[10 поена] Провера стања загонетке**

Током формирања решења загонетке могуће је проверити да ли је тренутно стање матрице валидно, односно да ли задовољава услове загонетке (нема понављања бројева).

**[30 поена] Формирање стабла исхода за матрицу  $4 \times 4$**

На основу читане почетне матрице  $4 \times 4$  треба формирати стабло одлучивања. **Све даље обраде се врше искључиво над формираним стаблом.** Чвор стабла одлучивања треба да представља једно стање матрице (иницијално, међустање или финално решење). Гране моделују унос одређеног броја у одређено поље матрице. Може се редуковати број чворова-потомака тако што се неће формирати чворови-потомци за стања која нису валидна. Листови стабла представљају или решења загонетке или стања из којих је закључено да не воде ка решењу.

**[50 поена] Обраде над формираним стаблом**

Након формирања стабла, програм треба да омогући кориснику следеће обраде:

**а) [20 поена] Испис стабла**

Омогућити испис изгледа формираног стабла (погодно форматиран, по нивоима).

У зависности од редног броја проблема који се решава  $i$ , потребно је реализовати решење ове ставке користећи један од следећа два начина обиласка:

1. *Preorder* обилазак стабла
2. *Level-order* обилазак стабла

**б) [10 поена] Испис свих решења у стаблу**

Коришћењем задатог обиласка исписати по корацима решење загонетке (ниједног, једног или више, уколико их има).

**в) [20 поена] Корисник решава игру**

Омогућити кориснику да уноси бројеве у матрицу „Судоку“ загонетке и на тај начин игра игру. Кориснику у сваком тренутку треба омогућити да може да провери да ли је до сада исправно унео бројеве (да ли је тренутно стање матрице исправно), да ли је „на добром путу“, односно да ли га тренутно стање води до решења, може да тражи помоћ (да му се предложи унос броја који води ка решењу), може да каже да нема решења или да одустане (крај игре).

Корисник са програмом интерагује путем једноставног менија. Програм треба да испише садржај менија, а затим да чека да корисник изабере (унесе путем тастатуре) редни број неке од понуђених ставки, након чега, пре извршења, од корисника очекује да по потреби унесе додатне параметре. Поступак се понавља све док корисник у менију не изабере опцију за прекид програма.