13С111АСП1 – ДЗ2 Рок за израду: 16.04.2018.

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1
2017-2018
- домаћи задатак 2 -

Опште напомене:

1. 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**.
2. Домаћи задатак 2 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику Pascal или C, по избору.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Решење треба да буде отпорно на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
5. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Коришћење рекурзије у решењима повлачи значајно умањење броја поена. Коришћење готових структура података није дозвољено**.
6. 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.
7. Одбрана другог домаћег задатка ће се обавити у **понедељак, 16.04.2018. и уторак, 17.04.2018.** према распореду који ће благовремено бити објављен на сајту предмета.
8. Формула за редни број комбинације проблема **i** и **j** који треба решавати је следећа: (R – редни број индекса, G – последње две цифре године уписа):

$$\mathbf{i = (R + G) \bmod 4 + 1}$$

$$\mathbf{j = (R + G) \bmod 2 + 1}$$

9. Име датотеке која се предаје мора бити **dz2p1.(pas|c)**.
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

03.04.2018. године *Са предмета*

Задатак – стабло претраге (100 поена)

Имплементација стабла (50 поена)

[30 поена] Написати програм који илуструје рад са стаблом целих бројева реда m ($m \geq 2$). Програм треба да омогући стварање стабла задатог степена, уметање новог елемента, испис стабла, као и брисање стабла. Уметање новог елемента извршити тако да стабло увек буде комплетно или скоро комплетно.

[15 поена] Зависно од дефинисаног редног броја **i** имплементирати и једну од следећих операција:

1. Одређивање висине стабла
2. Одређивање ширине стабла
3. Одређивање броја чворова стабла чији је степен једнак реду стабла
4. Одређивање броја чворова стабла

Све наведене операције треба реализовати путем одговарајућих потпрограма чији је један од аргумената стабло над којим се врше операције. Зависно од редног броја **j** потребне обиласке у стаблу имплементирати **искључиво** коришћењем:

1. *preorder* алгоритма
2. *level order* алгоритма

[5 поена] 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.

Имплементација игре (50 поена)

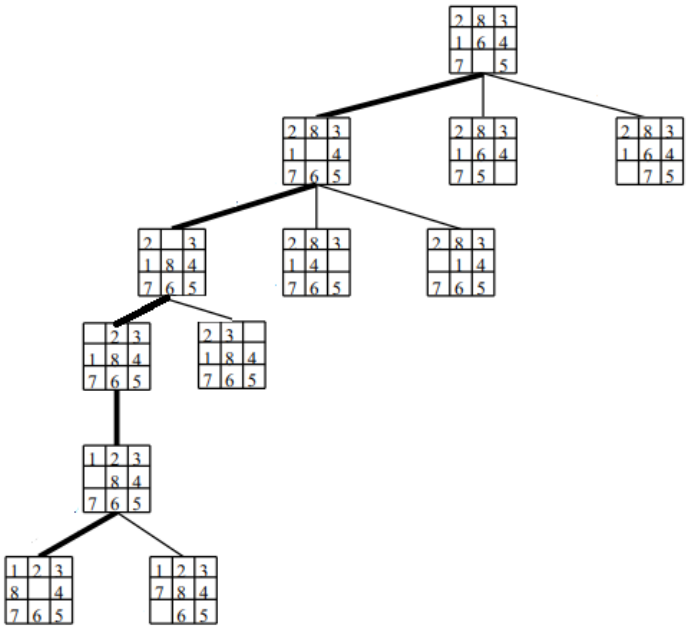
Модификовањем претходно написаног решења, написати програм за решавање познате дечије слагалице у којој се на табли димензија 3x3 налази низ плочица које су означене бројевима од 1 до 8, као на слици 1. Једно поље табле је празно, односно није покривено плочицом. У сваком потезу могуће је померити једну од плочица са бројем које су суседне празном пољу, на место празног поља. Циљ игре је сложити плочице у неки карактеристичан распоред који је задао корисник.

2	8	3
1	6	4
7		5

1	2	3
8		4
7	6	5

Слика 1. Пример почетне и крајње позиције

Програм треба да формира стабло претраге и да помоћу њега прикаже једно од могућих решења. Сваки чвор стабла представља стање игре – распоред плочица на табли у одговарајућем потезу. Корену стабла одговара задата почетна позиција, док се крајња позиција налази у листу стабла. Одабрано решење приказати исписивањем изгледа табле након сваког потеза. У случају да не постоји решење за дате улазне параметре, програм треба да испише одговарајуће обавештење. Пример стабла претраге које одговара почетној и крајњој позицији са слике 1. дат је на слици 2.



Слика 2. Пример стабла претраге задате игре

[5 поена] 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. За практичну примену, корисник програма треба да има барем следеће могућности реализоване путем одговарајућих ставки менија:

1. **[5 поена]** Уношење почетне и крајње позиције са стандардног улаза
2. **[5 поена]** Генерисање почетне и крајње позиције попуњавањем псеудослучајним бројевима
3. **[10 поена]** Провера да ли је циљна позиција достижна из почетне
4. **[25 поена]** Симулација игре

Почетна и крајња позиција задају се дефинисањем распореда плочица на табли у виду матрице уносом са стандардног улаза или насумичним попуњавањем. За генерисање почетне и крајње позиције се може искористити *Fisher-Yates* алгоритам за генерисање случајне пермутације низа бројева. Пример задате почетне и крајње позиције дат је на слици 1 и слици 3.

Провера да ли је проблем решив за произвољну величину поља може се извршити на следећи начин:

1. Из почетне позиције, уз поштовање правила игре, доћи у нову позицију у којој се празно поље налази на истом месту као и у циљној позицији.
2. Свако место у новој позицији, почев од горњег левог угла закључно са доњим десним, обрадити на следећи начин: ако број А на том месту није једнак броју Б на истом месту у циљној позицији, заменити у посматраној позицији места бројевима А и Б (замена је директна, не по правилима игре).
3. Ако је број замена у тачки 2. био паран, позиције су у истој класи парности. У супротном, позиције нису у истој класи и задатак нема решења.

На слици 3 је приказан пример нерешивог проблема. До крајње позиције се не може доћи из дате почетне позиције, јер је број потребних замена у тачки 2. једнак јединици – позиције нису у истој класи парности.

1	2	3
4	5	6
7	8	

1	2	3
4	5	6
8	7	

Слика 3. Пример нерешивог проблема

Симулација игре подразумева испис решења по потезима, ако је проблем решив, уз приказ изгледа табле након сваког одиграног потеза, од почетне до крајње позиције. До решења се долази формирањем стабла претраге за задату почетну и крајњу позицију, као на слици 2. На слици су подебљане гране стабла које воде до циљне позиције. Довољно је приказати једно од могућих решења.

Потребне обиласке у стаблу имплементирати **искључиво** коришћењем алгоритма за обилазак који је дефинисан у првом делу задатка.

Више информација о проблему се може видети на следећим страницама:

https://en.wikipedia.org/wiki/Sliding_puzzle
https://en.wikipedia.org/wiki/15_puzzle
https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle