

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2016-2017
- први домаћи задатак -

Опште напомене:

1. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику С или С++.
2. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
3. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
4. Решења треба да буду отпорна на грешке и треба да кориснику пруже јасно обавештење у случају детекције грешке.
5. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као успешно решење проблема.**
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 низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
7. Одбрана трећег домаћег задатка ће се обавити у **среду, 26.10.2016. и четвртак, 27.10.2016.** према распореду који ће накнадно бити објављен на сајту предмета.
8. Пре одбране, сви студенти раде тест знања за рачунаром коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да се пријаве на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико је студент регистрован на систем путем свог налога електронске поште на серверу mail.student.etf.bg.ac.rs.
9. Други задатак је исти за све студенте, а формула за редни број **i** комбинације проблема коју треба користити приликом решавања првог задатка је следећа:
(**R** – редни број индекса, **G** – последње две цифре године уписа):
$$i = (R + G) \bmod 4$$
10. Имена датотека које се предају морају бити **dz1p1.c** и **dz1p2.c**
11. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

18.10.2016. године

Са предмета

Задатак 1 – стабло бинарног претраживања [60 поена]

Написати програм за рад са стаблом бинарног претраживања. Чвор стабла треба да садржи **само** целобројни кључ и показиваче на лево и десно подстабло. У стаблу је дозвољено смештање кључева исте вредности. Ради олакшавања операција са стаблом, користи се посебан **чвор-граничник**. Над стаблом бинарног претраживања је могуће извршити следеће операције:

- a) Претрагу стабла на произвољан кључ
- b) Проналажење претходника задатог кључа
- c) Проналажење следбеника задатог кључа
- d) Уметање кључа у стабло; уколико кључ постоји, умеће се као претходник
- e) Уметање кључа у стабло; уколико кључ постоји, умеће се као следбеник
- f) Брисање кључа из стабла уз коришћење претходника
- g) Брисање кључа из стабла уз коришћење следбеника
- h) Утврђивање да ли задато стабло представља стабло бинарног претраживања
- i) Инвертовање стабла (тако да се примени алтернативни критеријум)
- j) Одређивање баланса задатог чвора према AVL критеријуму

У зависности од редног броја проблема који се решава, реализовати следећу комбинацију операција:

0. a), b), d), f), h)
1. a), c), e), g), j)
2. a), b), e), f), i)
3. a), c), d), g), i)

Демонстрацију рада програма остварити кроз једноставан интерактивни мени који ће омогућити комуникацију са корисником. Програм треба да чита скуп целобројних кључева који се умећу у стабло са стандардног улаза или из датотеке и да има могућности исцртавања изгледа стабла. Омогућити у менију и стварање празног стабла, брисање стабла и уметање и брисање појединачних кључева.

Задатак 2 – претраживање линеарних структура података [40 поена]

Нека је дата матрица димензија $M \times N$ која садржи целобројне кључеве. Кључеви могу да се понављају. Познато је да су врсте и колоне матрице уређене неопадајуће или нерастуће, али се у општем случају линеаризацијом матрице по врстама или колонама не добија уређени низ. Начин уређености врста и колона матрице је познат унапред. Осмислити и имплементирати функцију за **ефикасну** претрагу овакве структуре података на задати кључ. Програм треба да омогући унос или генерисање матрице са описаним карактеристикама. Евалуацију перформанси претраживања треба омогућити генерисањем секвенце кључева и одређивањем просечног броја корака потребног за претраживање.

Пример матрице 5x4 чије су врсте и колоне уређене неопадајуће

10	15	20	25
25	42	45	80
36	56	75	90
46	75	95	115
57	89	111	139