

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2022-2023

- први домаћи задатак -

Опште напомене:

1. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику C или C++.
2. 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.
3. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
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 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 низа). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. За решавање задатака који имају више комбинација користити следеће формуле. (**R** – редни број индекса, **G** – последње две цифре године уписа):
$$i = (R + G) \bmod 4 + 1$$
$$j = (R + G) \bmod 3 + 1$$
9. Предаја домаћег задатка ће бити омогућена преко Moodle система. Детаљније информације ће бити благовремено објављене.
10. Одбрана првог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
11. 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 1 – претраживање линеарних структура података [60 поена]

Нека је дат уређени низ K дужине k који садржи целобројне кључеве. У зависности од редног броја i проблема који се решава одређују се карактеристике низа који је потребно имплементирати (да ли се кључеви могу понављати и уређеност низа):

1. Неоппадајући низ без могућег понављања кључева
2. Неоппадајући низ са могућим понављањем кључева
3. Нерастући низ без могућег понављања кључева
4. Нерастући низ са могућим понављањем кључева

Потребно је имплементирати m -арну претрагу, која функционише по принципу уопштене бинарне претраге тако што низ уређених кључева дели на m приближно једнаких делова и претрагу наставља у једном од њих. Процес поделе се понавља у изабраном делу док се тражени кључ не пронађе или се не утврди да се он не налази у низу. Уколико параметар m има вредност два, претрага се своди на класичну бинарну претрагу, уколико узима вредност три, претрага се своди на тернарну претрагу, итд.

Програм треба да омогући следеће функционалности:

1. **[5 поена]** Унос уређеног низа кључева K дужине k из датотеке или са стандардног улаза (корисник бира одакле се уноси).
2. **[10 поена]** Генерисање уређеног низа кључева K величине k у задатом опсегу, где параметар k и опсег задаје корисник. Вредности кључева треба да буду **псеудослучајни** бројеви из задатог опсега, а да се при том очува уређеност низа.
3. **[20 поена]** Претрагу уређеног низа K на задати кључ употребом m -арне претраге. Вредност параметра m и кључа по коме се врши претрага задаје корисник.
4. **[15 поена]** Оптимизовану претрагу уређеног низа K на низ кључева P употребом m -арне претраге. Корисник уноси уређени низ кључева P дужине p на које се врши претрага. Кључеви су уређени на исти начин као и низ над којим се ради претрага. Параметре m и p задаје корисник.
5. **[5 поена]** Евалуацију перформанси m -арног претраживања генерисањем секвенце кључева у задатом опсегу и одређивањем просечног броја корака потребног за претраживање. Евалуацију вршити за све вредности параметра m из скупа вредности [2-6] и приказати добијене резултате за сваку вредност параметра. Кључеви на које се врши претрага задају се један по један.
6. **[5 поена]** Евалуацију перформанси оптимизованог m -арног претраживања над низом кључева генерисањем секвенце кључева у задатом опсегу и одређивањем просечног броја корака потребног за претраживање. Евалуацију вршити за све вредности параметра m из скупа вредности [2-6] и приказати добијене резултате за сваку вредност параметра.

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.

Задатак 2 – имплементација BST са понављањем кључева [40 поена]

Написати програм за рад са бинарним стаблом претраживања (BST) које садржи целобројне кључеве. BST се креира на основу уређеног низа кључева K дужине k из задатка 1. Сматрати да у оквиру чвора стабла постоји показивач лево и десно подстабло, као и показивач на оца. Уређеност кључева у стаблу може се модификовати тако да буде у складу са уређеношћу низа K .

Потребно је подржати следеће операције за рад са стаблом бинарног претраживања:

1. [10 поена] Формирања стабла на основу уређеног низа кључева K
2. [10 поена] Уметање новог кључа у стабло
3. [5 поена] Претрагу стабла на задати кључ
4. [5 поена] Форматиран испис садржаја стабла
5. [5 поена] Евалуацију перформанси претраживања у BST над низом кључева генерисањем секвенце кључева у задатом опсегу и одређивањем просечног броја корака потребног за претраживање.
6. [5 поена] Брисање стабла из меморије

У зависности од редног броја j проблема који се решава, уметање поновљених кључева реализовати на следећи начин:

1. Чувањем листе свих кључева у оквиру једног чвора стабла
2. Уметањем поновљеног кључа као **следбеника** чвора истог кључа
3. Уметањем поновљеног кључа као **претходника** чвора истог кључа

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.

Напомена: Код исписа стабла, испис неке комбинације обилазака стабла се не прихвата као адекватно решење, већ стабло треба адекватно форматирати.