

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

- други домаћи задатак -

Опште напомене:

1. Домаћи задатак 2 састоји се од једног програмског 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. Предаја домаћег задатка ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене.
9. Одбрана домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
10. 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.

18.11.2022. године

Са њредмета

Задатак 1 – Ефикасно претраживање табеле базе података [100 поена]

Нека је дата табела *CUSTOMER_ACCOUNT* која представља табелу из једног стандардног *benchmark*-а за тестирање перформанси база података. Структура ове табеле је следећа: (*CA_ID*, *CA_B_ID*, *CA_C_ID*, *CA_NAME*, *CA_TAX_ST*, *CA_BAL*). *CA_ID* је примарни кључ табеле (целобројни податак), и индекс се креира над њим. Параметри *CA_B_ID*, *CA_C_ID*, *CA_TAX_ST* су административни подаци везани за рачун (целобројног типа), *CA_NAME* представља име рачуна (string), а *CA_BAL* представља стање на рачуну (реалан број).

Табела је дата текстуалном датотеком која има следећу структуру (| представља делимитер у датотеци):

CA_ID|CA_B_ID|CA_C_ID|CA_NAME|CA_TAX_ST|CA_BAL

Потребно је имплементирати Б+ стабло реда m (где је m минимално 3, а максимално 10) као индексну структуру (у наставку индекс) над табелом *CUSTOMER_ACCOUNT* ради ефикасног претраживања рачуна по примарном кључу. За сваки кључ је везан и показивач на структуру која чува све податке о рачуну.

Реализовати следеће операције над индексом:

1. **[20 поена]** Стварање и уништавање индекса над табелом *CUSTOMER_ACCOUNT*. Приликом стварања индекса корисник уноси ред Б+ стабла.
2. **[5 поена]** Испис индекса на стандардни излаз тако да су појединачни чворови уочљиви (међусобно размакнути). Ова ставка ће бити тестирана за мале величине табела (репрезентативна величина табеле за ову ставку је дата у датотеци *CustomerAccount20.txt*)
3. **[15 поена]** Додавање новог записа (рекорда) у табелу (уносе се сви подаци о рачуну)
4. **[15 поена]** Брисање записа у табели по задатом примарном кључу. Корисник задаје примарни кључ записа који се брише.
5. **[15 поена]** Претраживање једног података. Корисник задаје један примарни кључ по коме се врши претрага. Претрага треба да на стандардни излаз испише одговарајући запис, као и укупан број корака за које је целокупна претрага завршена.
6. **[15 поена]** Претраживање k података. Корисник задаје број примарних кључева (параметар k) и низ кључева дужине k над којима се врши претрага. Кључеви у низу су у произвољном редоследу. Претрага треба да на стандардни излаз испише укупан број корака за које је целокупна претрага завршена, као и да креира датотеку са пронађеним записима, према горњем формату.
7. **[15 поена]** Претраживање k сукцесивних података. Корисник задаје број примарних кључева (параметар k) и примарни кључ почетног записа. Претрага треба да на стандардни излаз испише укупан број корака за које је целокупна претрага завршена, као и да креира датотеку са пронађеним записима, према горњем формату.

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.

Напомене

По потреби реализовати и додатне методе, где је то примерено.

За тестирање програма се могу користити датотеке са табелама различитих величина, које се налази у оквиру посебне архиве.

Рад са датотекама у језику C++ захтева увожење заглавља `fstream` (именски простор `std`). За читање података користи се класа `ifstream`. Након отварања датотеке, читање се врши на исти начин као и са стандардног улаза. Кратак преглед најбитнијих метода и пријатељских функција ове класе је дат у наставку.

<pre>void open(const char *_Filename, ios_base::openmode _Mode = ios_base::in, int _Prot = (int)ios_base::_Openprot);</pre>	Отвара датотеку задатог имена за читање. <code>ifstream dat;</code> <code>dat.open("datoteka.txt");</code>
<pre>void close();</pre>	Затвара датотеку.
<pre>bool is_open();</pre>	Утврђује да ли је датотека отворена.
<pre>operator>></pre>	Преклопљен оператор за просте типове података.
<pre>ifstream dat; dat.open("datoteka.dat"); if(! dat.is_open()) greska(); char niz[20]; dat >> niz; dat.close();</pre>	Пример отварања датотеке, провере да ли је отварање успешно, читање једног знаковног низа из датотеке и затварања датотеке.