

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2024-2025
- други домаћи задатак -

Опште напомене:

1. Домаћи задатак 2 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C или C++.
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 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 успешно решење проблема које може освојити максималан број поена.**
7. Дозвољено је коришћење линеарних структура података из стандардне библиотеке *stl*.
8. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити које претпоставке и која ограничења су усвојена (на пример, максимална димензија низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
9. Одбрана првог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
10. За решавање задатака који имају више комбинација користити следеће формуле.
(**R** – редни број индекса, **G** – последње две цифре године уписа):

$$i = (R + G) \bmod 2$$

$$j = (R + G) \bmod 3$$

11. Предаја домаћих ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене. Име архиве која се предаје мора бити **asp2dz2.zip**. Погрешно назван и предат домаћи задатак повлачи одузимање 25 поена. Закаснила предаја домаћег задатка неће бити омогућена. Није потребно предавати материјале са сајта предмета приложене уз домаћи задатак.
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. Током израде решења није дозвољена употреба алата вештачке интелигенције заснованих на великим језичким моделима (*ChatGPT*, *Github Copilot* и сл.).

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

Нека је дата табела *CUSTOMER* која представља табелу из једног стандардног *benchmark*-а за тестирање перформанси база података. Структура ове табеле је следећа: (*C_ID*, *C_F_NAME*, *C_L_NAME*, *C_EMAIL*, *C_AD_ID*). *C_ID* је примарни кључ табеле (целобројни податак). Структура података која се користи за убрзавање приступа подацима се назива индекс. Индекс се креира над примарним кључем ове табеле. Параметри *C_F_NAME*, *C_L_NAME*, *C_EMAIL_1*, *C_AD_ID* су подаци везани за клијента (име, презиме, e-mail и страни кључ на другу табелу у којој се чувају адресе).

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

C_ID|*C_F_NAME*|*C_L_NAME*|*C_EMAIL*|*C_AD_ID*

Потребно је имплементирати **Б*** стабло реда m (где је m минимално 3, а максимално 10) као индексну структуру (у наставку индекс) над табелом *CUSTOMER* ради ефикасног претраживања корисника по примарном кључу. Имплементација информационог садржаја појединачног чвора стабла се оставља студенту на имплементацију и биће предмет оцењивања на самој одбрани.

Реализовати следеће операције над индексом:

1. **[20 поена]** Стварање и уништавање индекса над табелом *CUSTOMER*. Приликом стварања индекса корисник уноси ред **Б*** стабла.
2. **[5 поена]** Испис индекса на стандардни излаз. Ова ставка ће бити тестирана за мале величине табела (репрезентативна величина табеле за ову ставку је дата у датотеци *Customer20.txt*)
3. **[15 поена]** Додавање новог записа (рекорда, реда) у табелу (уносе се сви подаци о кориснику)
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 који је целокупна претрага завршена.
6. **[15 поена]** Претраживање свих података на име корисника (поређење са атрибутом табеле *C_F_NAME*). Претрага треба да на стандардни излаз испише одговарајуће записе, као и укупан број корака за који је целокупна претрага завршена. Ову претрагу реализовати најефикасније могуће.
7. **[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 који 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.