

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

Опште напомене:

1. Домаћи задатак 2 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику C++.
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 другог домаћег задатка ће се обавити у **среду, 30.11.2016. и четвртак, 01.12.2016.** 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. Имена датотека које се предају морају бити **dz2p1.cpp** и **dz2p2.cpp**
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

22.11.2016. године

Са предмета

Задатак 1 – Б стабло [70 поена]

Нека је дата структура података коју чине стрингови. Структура података се састоји од динамичког низа стрингова и помоћног низа који чува податке о дужини за сваки од стрингова. Стрингови у низу су сортирани по дужини, а стрингови исте дужине су смештени у лексикографском поретку.

Како би се олакшало и убрзало претраживање овакве структуре података према дужини стрингова, потребно је креирати **редак** индекс коришћењем Б стабла произвољног реда. У једном чвору Б стабла се чувају целобројни кључеви који представљају дужине стрингова који се налазе у структури података. За сваки кључ је везана адреса првог стринга у структури података са одговарајућом дужином.

Реализовати следеће операције:

1. [10 поена] стварање и основна манипулација структуром података која садржи стрингове
 - стварање на основу података из задате датотеке или са стандардног улаза
 - уметање стринга
 - брисање стринга
 - уништавање
 - исписивање садржаја структуре у излазни ток (оператор <<)
2. [20 поена] стварање и основна манипулација ретким индексом над структуром података која садржи стрингове
 - стварање објекта стабла **задатог реда**
 - брисање објекта стабла
 - одређивање висине стабла
 - одређивање броја кључева уметнутих у стабло
 - проналажење кључа у стаблу (враћа вредност `true` ако кључ постоји, односно `false` у супротном)
 - исписивање садржаја стабла у излазни ток (оператор <<), по нивоима, тако да су поједини чворови уочљиви (међусобно размакнути)
3. [20 поена] динамичка манипулација ретким индексом
 - уметање кључа у стабло (враћа вредност `false` ако кључ већ постоји, односно `true` у случају успешног уметања)
 - брисање кључа из стабла (враћа вредност `false` ако кључ не постоји, односно `true` у случају успешног брисања)
4. [20 поена] динамичка манипулација структуром података која садржи стрингове уз одржавање ретког индекса
 - проналажење стринга
 - уметање стринга
 - брисање стринга
 - исписивање свих стрингова са задатом дужином

Обратити пажњу да се након уметања новог стринга у структуру података мора извршити ажурирање ретког индекса.

Задатак 2 – црвено-црна стабла [30 поена]

Познато је да постоји изоморфизам између 2-3-4 стабала и црвено-црних стабала, а да су 2-3-4 стабла варијанта Б стабла реда 4. Модификовати имплементацију Б стабла из претходног задатка тако да подржи основне операције са црвено-црним стаблом. Написати методу која задато Б стабло трансформише у црвено-црно стабло и омогућава његов испис на екран. Имплементација треба да омогући операције претраге и уметања кључа у црвено-црно стабло коришћењем модификованог Б стабла.

Напомене

У оба задатка, по потреби реализовати и додатне методе, где је то примерено. Написати главне програме који кориснику омогућавају рад са задатим стаблом, путем једноставног интерактивног менија.

Рад са датотекама у језику C++ захтева увођење 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.