

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1
2021-2022

- први домаћи задатак -

Опште напомене:

1. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблем решавају **самостално**, на програмском језику С или Python.
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 путем читања са стандардног улаза.
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 и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Одбрана првог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
9. За решавање задатака који имају више комбинација користити следеће формуле.
(**R** – редни број индекса, **G** – последње две цифре године уписа):

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

$$j = (R + G) \bmod 4$$

10. Имена датотека које се предају мора бити **dz1p1.(c|py)** и **dz1p2.(c|py)**
11. Предаја домаћих ће бити омогућена преко Moodle система. Детаљније информације ће бити благовремено објављене.
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.

Задатак 1 – имплементација уланчаних листа [50 поена]

[45 поена] Написати интерактиван програм који коришћењем одговарајућег типа уланчаних листа прати информације о положеним предметима за студенте Електротехничког факултета.

Нека се посматра имплементација информационог система о студентима Електротехничког факултета. Ради поједностављења проблема, за сваког студента се памти само индекс и листа положених предмета. При додавању новог студента у систем, листа положених предмета се ствара празна. Сваки предмет у листи садржи следеће две информације: шифру предмета и оцену. Листу положених предмета је потребно одржавати уређено лексикографски по шифри предмета. У даљем раду програма, потребно је омогућити кориснику да за одређеног студента дода положен предмет у листу, поништи (обрише) предмет, филтрира предмете са одређеним условом, испише све положене предмете и дохвати број положених предмета. Обратите пажњу да два различита студента не смеју имати идентичан индекс, као и да два различита предмета не смеју имати идентичну шифру. Чување конкретних студената је дозвољено имплементирати у виду уланчане листе, као и у виду статичког низа. Уколико се користи друга варијанта, потребно је кориснику омогућити да проследи максималан број студената у структури.

Зависно од редног броја проблема i , саставити **један** од следећих програма, који реализују:

0. Унију предмета који су положили два задата студента
1. Пресек предмета који су положили два задата студента
2. Разлику предмета који су положили два задата студента

Зависно од редног броја проблема j , саставити **један** од следећих програма у којем је листа предмета реализована као:

0. Двоструко уланчана листа са заглављем
1. Двоструко уланчана кружна листа са заглављем
2. Једноструко уланчана листа са заглављем
3. Једноструко уланчана кружна листа са заглављем

[5 поена] 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. За практичну примену, корисник програма треба да има најмање следеће могућности реализоване путем одговарајућих ставки менија:

1. Стварање новог студента
2. Додавање положеног предмета за одређеног студента
3. Приказ свих предмета са одређеном оценом
4. Поништавање (брисање) положеног предмета за одређеног студента
5. Брисање целе листе положених предмета за одређеног студента
6. Дохватање броја положених предмета за студента
7. Исписивање свих положених предмета одређеног студента

Задатак 2 – алгоритми за рад са ретким матрицама (50 поена)

Написати интерактиван програм на програмском језику C или Python који илуструје рад са ретким матрицама. Ретке матрице су врста матрица код којих већина елемената има подразумевану вредност. Код оваквих матрица се могу направити значајне уштеде у простору потребном за њихово складиштење уколико се експлицитно памте само вредности које нису подразумеване. У наставку је дат кратак опис три технике смештања ретких матрица.

Вектор записа од три поља (Coordinate Of Objects – COO) је начин смештања ретке матрице димензија $M \times N$ код кога се подразумеване вредности памте експлицитним чувањем тројки које чине врсту, колону и вредност елемента. Тројке у вектору су уређене најпре по редном броју врсте, а затим по редном броју колоне.

Compressed row storage (CSR) формат чува ретку матрицу димензија $M \times N$ помоћу три независна вектора (V, C, R). Вектор V садржи вредности и има онолико елемената колико има подразумеваних елемената матрице. Вектор C садржи број колоне одговарајућег елемента вектора V . Сваки елемент вектора R одговара једној врсти матрице и садржи индекс првог елемента вектора C и V који одговара тој врсти. Вектор R има дужину $M+1$, а у последњем елементу се чува укупан број подразумеваних елемената матрице.

Compressed sparse column (CSC) формат чува ретку матрицу димензија $M \times N$ на сличан начин као CSR формат, осим што вектори R и C замењују места и значење. Вектор V садржи вредности и има онолико елемената колико има подразумеваних елемената матрице. Вектор R садржи број врсте одговарајућег елемента вектора V . Сваки елемент вектора C одговара једној колони матрице и садржи индекс првог елемента вектора R и V који одговара тој колони. Вектор C има дужину $N+1$, а у последњем елементу се чува укупан број подразумеваних елемената матрице.

Зависно од редног броја проблема i , саставити **један** од следећих програма, који представља ретку матрицу помоћу:

0. једног вектора записа од три поља
1. три независна вектора у CSR формату
2. три независна вектора у CSC формату

Програм треба да омогући стварање матрице и унос ретке матрице задатих димензија и задатог броја ненултих елемената, дохватање вредности елемента задате врсте и колоне, испис матрице (укључујући и елементе подразумеване вредности). Посебно, зависно од редног броја проблема j , саставити **један** од следећих потпрограма који омогућавају следећу операцију за рад са ретком матрицом:

0. сабирање две матрице уз проверу да ли се могу сабрати
1. транспоновање матрице
2. сума подматрице која је одређена координатама горње левог и доње десног темена
3. ротирање матрице за 90° у смеру казаљке на сату

Напомена: у матрици је потребно чувати само подразумеване елементе. Више информација о наведеним техникама се може добити на следећој адреси: https://en.wikipedia.org/wiki/Sparse_matrix.