

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1

2018-2019

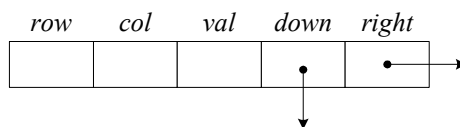
- домаћи задатак 1 -

Опште напомене:

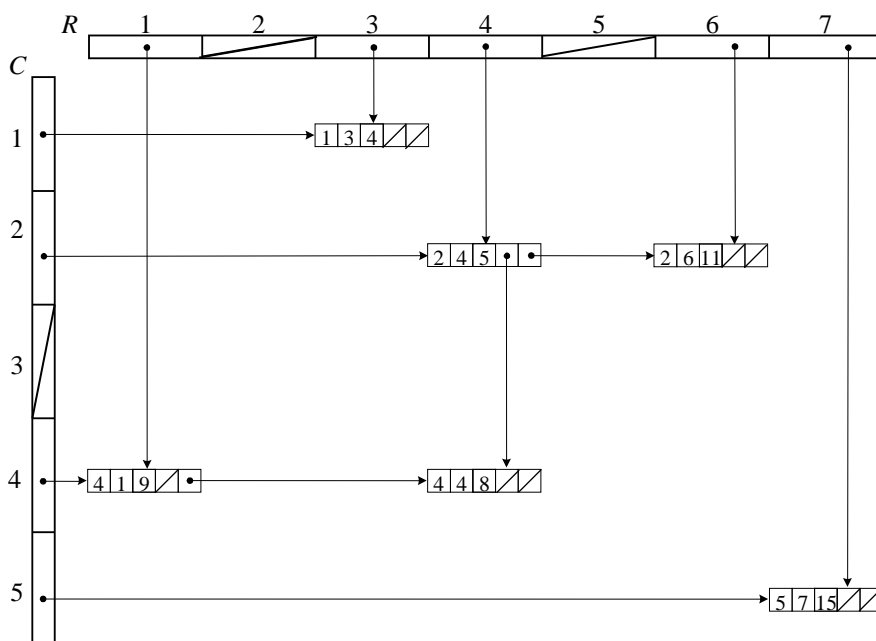
1. Пре одбране сви студенти раде тест знања који се ради на рачунару коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да креирају налог и пријаве се на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити **прихваћена и важећа** само уколико се студент региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
2. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмским језицима Pascal или C, по избору.
3. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Решења треба да буду отпорна на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
5. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса.
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 матрице и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
7. Одбрана првог домаћег задатка ће се обавити у **понедељак, 11.03.2019. и уторак, 12.03.2019.** према распореду који ће благовремено бити објављен на сајту предмета.
8. Формула за редни број проблема **i** који треба решавати у **првом** задатку је следећа: (R – редни број индекса, G – последње две цифре године уписа): **$i = (R + G) \bmod 3 + 1$**
9. Формула за редни број проблема **i** који треба решавати у **другом** задатку је следећа: (R – редни број индекса, G – последње две цифре године уписа): **$i = (R + G) \bmod 4 + 1$**
10. Имена датотека које се предају морају бити **dz1p1.(pas|c)** и **dz1p2.(pas|c)**
11. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

Задатак 1 – представљање ретке матрице уланчаним листама (50 поена)

Написати интерактиван програм који илуструје рад са ретким матрицама за чију се репрезентацију користе једноструко уланчане листе и два низа R и C . Ретке матрице су оне матрице 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. Свака врста и колона представљене су по једном листом, чији чворови садрже неподразумеване елементе матрице. Чвор листе садржи следећа поља: row – индекс врсте елемента, col – индекс колоне, val – вредност посматраног елемента, $down$ – показивач на следећи неподразумевани елемент у истој колони, $right$ – показивач на следећи неподразумевани елемент у истој врсти. Један чвор се тако налази у две листе – у листи за колону col и листи за ред row . Низови $R[i]$ и $C[j]$ садрже показиваче на први неподразумевани елемент у i -тој врсти, односно, први неподразумевани елемент у j -тој колони. Листе нису кружне, те уколико се у посматраном реду или колони не налази ниједан елемент, у низу се налази празан показивач (празна листа). Величине низова R и C одређене су бројем врста и колоне за дату матрицу. Изглед чвора и пример матрице дати су на сликама 1 и 2.



Слика 1. Изглед чвора листе



Слика 2. Изглед једне ретке матрице представљене листама

Програм треба да омогући ефикасан приступ произвољном елементу матрице. 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 следеће могућности реализоване путем одговарајућих ставки менија:

1. стварање матрице задатих димензија уз иницијализацију не подразумеваним вредностима
2. постављање подразумеване вредности
3. дохватање задатог елемента, уз проверу валидности опсега
4. постављање вредности задатом елементу, уз проверу валидности опсега
5. дохватање броја неподразумеваних елемената
6. испис матрице (укључујући и елементе подразумеване вредности)
7. брисање матрице

Зависно од редног броја проблема, реализовати и **једну** од следећих операција којом се:

1. формира нова ретка матрица сабирањем две задате ретке матрице.
2. формира нова ретка матрица транспоновањем задате ретке матрице.
3. формирају једна горње троугаона и једна строго доње троугаона матрица које представљају збир задате ретке матрице.

Напомена: у матрици је потребно чувати само неподразумеване елементе, а структуру података треба ажурирати након сваке операције промене вредности неког елемента матрице.

Задатак 2 – векторска имплементација приоритетног реда (50 поена)

Написати програм који илуструје рад са приоритетним редовима представљеним у виду вектора. Елементи реда су цели бројеви. Мања вредност означава већи приоритет. Програм треба омогући стварање реда (са задатим максималним капацитетом), као и следеће основне операције: уметање елемента у ред, уклањање елемента из реда (уз враћање вредности), дохватање елемента са почетка реда (без уклањања), дохватање броја заузетих места у реду, дохватање максималног капацитета реда. Све наведене операције треба реализовати путем одговарајућих потпрограма чији је један од аргумената показивач на ред над којим се врше операције. 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.

Зависно од редног броја проблема, **уметање и уклањање** елемента из реда реализовати на **један** од следећих начина:

1. Користећи технику маркирања елемената приликом брисања, при чему се приликом уметања не дозвољава упис преко маркираног елемента; уметање се увек врши на крај реда уз евентуално сажимање.
2. Користећи технику маркирања елемената приликом брисања, при чему се приликом уметања нови елемент уписује преко првог маркираног елемента (ако такав постоји).
3. Померањем преосталих елемената реда за једно место наниже или навише приликом брисања, у зависности од позиције елемента који се брише; уметање је увек на крај реда.
4. Користећи модификовану технику кружног бафера уз одржавање уређености реда по логичком приоритету елемената.

Саму структуру података моделовати помоћу кориснички дефинисаног типа (записа) који ће поред линеарне структуре података (низа) употребљене за реализацију реда садржати и додатне податке о самој структури података (попут тренутне и максималне величине, показивача на почетак и крај реда и сл.). Више информација о структурама података које треба реализовати се може наћи у поглављу „Векторска имплементација приоритетног реда“ у књизи „Алгоритми и структуре података“ проф. др. Мила Томашевића.