ОДСЕК ЗА РАЧУНАРСКУ ТЕХНИКУ И ИНФОРМАТИКУ

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА

2019-2020

- домаћи задатак -

Опште напомене:

1. Домаћи задатак састоји се од два програмска проблема. Студенти проблем решавају **самостално**, на програмском језику C++. **Није дозвољено коришћење готових структура података из STL и сличних библиотека.**
2. Право да раде домаћи задатак имају сви студенти који прате предмет. Студенти треба да се пријаве за одбрану домаћег задатка најкасније до **среде, 11.12.2019. у 10 часова** путем одговарајућег линка на сајту предмета.
3. Домаћи задатак замењује други део испита, односно задатак на испиту који се пише у конкретном језику. Поени са домаћег задатка ће стога бити скалирани у одговарајућем односу. Одбраном домаћег задатка студент се одриче права да ради други део испита. Студент на испиту треба да назначи да ли је радио домаћи задатак како би му били признати поени.
4. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
5. Унос података треба омогућити путем читања са стандардног улаза.
6. Решења треба да буду отпорна на грешке и треба да кориснику пруже јасно обавештење у случају детекције грешке.
7. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као успешно решење проблема и неће бити оцењена са максималним бројем поена.**
8. 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.
9. Одбрана домаћег задатка ће се обавити у **петак, 13.12.2019.**
10. Формула за редни број комбинације проблема **i** и **j** који треба решавати је следећа (R – редни број индекса, G – последње две цифре године уписа):

$$\mathbf{i = (R + G) \bmod 3 + 1}$$

$$\mathbf{j = (R + G) \bmod 4 + 1}$$

11. Имена датотеке које се предаје мора бити **dzp1.cpp** и **dzp2.cpp**
12. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

03.12.2019. године *Са предмета*

Графови [100 поена]

Граф је нелинеарна структура података која се састоји од скупа чворова и скупа грана. Гране представљају односе (везе) између чворова. Графови се могу користити за моделирање произвољних нелинеарних релација. Постоје усмерени и неусмерени графови.

Репрезентација графа

У зависности од редног броја **i** добијеног коришћењем формуле назначене у напоменама, потребно је користити једну од следећих меморијских репрезентација графа приликом решавања задатих проблема:

1. Матричну репрезентацију коришћењем матрица суседности
2. Уланчану репрезентацију коришћењем листа суседности
3. Секвенцијалну репрезентацију коришћењем линеаризованих листа суседности

Више информација о наведеним меморијским репрезентацијама графа се може пронаћи у материјалима са предавања и вежби, као и у књизи проф. Мила Томашевића „Алгоритми и структуре података“.

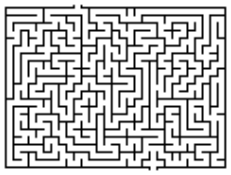
Задатак 1 - Имплементација основних алгоритама за рад са графом [50 поена]

Написати програм на програмском језику C++ који илуструје рад са неусмерним графовима. Програм треба да омогући следеће операције над графом:

- Креирање празне структуре података за граф задатих димензија (постоје чворови графа, али не постоје гране)
- Додавање чвора у граф и уклањање чвора из графа
- Додавање и уклањање гране између два чвора у графу
- Испис репрезентације графа
- Брисање графа из меморије

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. Све наведене операције треба реализовати путем одговарајућих потпрограма чији је један од аргумената показивач на структуру података која имплементира граф са којим се ради.

Задатак 2 - Проблем лавиринта [50 поена]



Коришћењем претходно написаних потпрограма за рад са графом, написати програм који симулира игру Лавиринт. Лавиринт има облик правоугаоника и ствара се задатих димензија. Сваки лавиринт има један улаз и један излаз који се морају наћи уз ивице лавиринта. Циљ игре је наћи путању од улаза до излаза. Пример лавиринта са једним улазом и једним излазом налази се на датој слици. Лавиринт може садржати слепе улице.

Лавиринт се може моделирати коришћењем неусмереног графа. Моделира се чворовима (ћелијама) који представљају „раскрснице“ – из сваког чвора се може прећи у суседни леви, десни, горњи или доњи чвор, под условом да између њих постоји грана, односно пролаз. Уколико грана не постоји, то значи да се између чворова налази зид и туда се не може проћи. Сваки чвор има свој идентификатор који се састоји од његових „координата“ у лавиринту (горњи-леви угао има координате (0,0), чвор десно од њега има координате (0,1), итд.). По потреби могу се додати и додатне информације потребне за решавање задатака.

[10 поена] Стварање и испис лавиринта

Корисник учитава податке о лавиринту са стандардног улаза. Најпре се уносе се подаци о димензијама лавиринта, позицији улаза и позицији излаза. На почетку се лавиринт може посматрати као матрица неповезаних чворова између којих се налазе зидови. Корисник затим може да уноси координате пролаза и тиме ствара путеве у лавиринту.

Треба омогућити испис унетог лавиринта на стандардни излаз или у текстуалну датотеку. Испис лавиринта на стандардни излаз ограничити само на лавиринт до максималних димензија 80x50, док за испис у датотеку такво ограничење не постоји. За исписивање зидова лавиринта се могу искористити знаци из основне или проширене ASCII табеле, нпр. коришћењем знака *, или на неки други произвољан начин. Пролазе треба исписати помоћу бланко знакова (размака), како би се видело куда може да се креће кроз лавиринт. Такође, улаз и излазе лавиринта треба обележити на неки начин (нпр. „o“ за улаз, „x“ за излазе).

[35 поена] Решавање лавиринта

Потребно је реализовати одговарајући алгоритам за решавање лавиринта. Решавање лавиринта подразумева налажење путање од улаза до излаза. Путања не мора да постоји и у том случају треба исписати одговарајућу поруку на стандардни излаз. Ако постоји решење, треба исписати низ корака које треба предузети да би се до излаза дошло. Низ корака се исписује у засебном реду, у формату: *(x0, y0) -> (x1, y1) -> (x2, y2) -> ... -> (xK, yK)*, где су *x0* и *y0* координате почетног чвора, *xK* и *yK* координате крајњег чвора, а *x1, y1, x2, y2* итд. координате чворова на пронађеној путањи..

У зависности од редног броја проблема **j с**тудент треба да реализује **један** од следећих алгоритама за решавање лавиринта:

1. Обилазак по ширини:
    a. Кренути од улазног чвора – текући чвор
    b. Обележити чвор као посећен
    c. Посетити суседе посећеног чвора у складу са логиком обиласка по ширини
    d. Понављати поступак док год има непосећених чворова или док се не посети чвор излаза
2. Дијкстрин алгоритам:
    a. Формирати иницијални вектор достижности на основу информација о суседности чворова
    b. Кренути од улазног чвора
    c. Користећи Дијкстрин алгоритам ажурирати вектор достижности
    d. Алгоритам може завршити своје извршавање када се дође до чвора који представља излаз
    e. Проверити да ли је чвор излаз достижан из полазног чвора

3. Обилазак по дубини:
    a. Кренути од чвора улаза – текући чвор је улазни
    b. Обележити чвор као посећен
    c. Посетити суседе посећеног чвора у складу са логиком обиласка по дубини
    d. Понављати поступак док год има непосећених чворова или док се не посети чвор излаза
4. Флојд-Варшалов алгоритам:
    a. Формирати иницијалну матрицу достижности на основу информација о суседности чворова
    b. Користећи Флојд-Варшалов алгоритам одредити матрицу достижности
    c. Проверити да ли је чвор излаз достижан из полазног чвора

[5 поена] Комуникација са корисником

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, уз обавезно обавештење шта се од корисника очекује да унесе. Поступак се понавља све док корисник у менију не изабере опцију за прекид програма. Све наведене операције треба реализовати путем одговарајућих потпрограма чији је један од аргумената показивач на структуру података која имплементира граф са којим се ради.