

**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1
2017-2018****- домаћи задатак 3 -****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, уколико то већ нису учинили. 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 3 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C.
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. Решења која употребљавају рекурзију не могу добити максималан број поена.
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, 28.05.2018.** и **уторак, 29.05.2018.** према распореду који ће бити објављен на сајту предмета.
8. Формула за редни број алгоритама **i** које треба користити приликом решавања задатка је следећа: (R – редни број индекса, G – последње две цифре године уписа):
$$i = (R + G) \bmod 3$$
9. Име датотеке која се предају мора бити **dz3p1.c**
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

18.05.2018. године

Са предмета

О графовима

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.

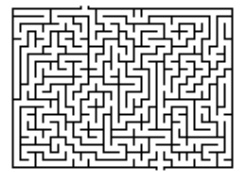
Репрезентација графа

Граф се може репрезентовати коришћењем матрица суседности или листа суседности. У оквиру овог задатка, студенти имају слободу да изаберу одговарајућу репрезентацију графа, у складу са потребама алгоритама које треба да имплементирају.

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

Задатак - Лавиринт (100 поена)

Написати програм на програмском језику C који симулира игру Лавиринт. Лавиринт има облик правоугаоника и ствара се задатих димензија. Сваки лавиринт има један улаз и више могућих излаза, који се морају наћи уз ивице лавиринта. Циљ игре је наћи путању од улаза до било ког излаза. Пример лавиринта са једним улазом и једним излазом налази се на датој слици. Лавиринт треба да буде без циклуса („петљи“) и 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 задатка.

[40 поена] Основне операције над графом

Програм треба да садржи и користи помоћне функције за стварање графа, додавање чворова, додавање грана, брисање чворова, брисање грана и брисање целог графа. У складу са потребама задатка, усвојити и имплементирати структуру података за репрезентацију графа над којом ће бити спровођене задате операције.

[25 поена] Стварање и испис лавиринта

Корисник може дефинисати број излаза из лавиринта, као и координате (индентификаторе) улаза и излаза. На почетку се лавиринт може посматрати као матрица неповезаних чворова између којих се налазе зидови. Стварање лавиринта се може обавити уношењем координата пролаза или генерисањем лавиринта помоћу једног од алгорита који су описани у наставку. Треба омогућити испис генерисаног 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. „о“ за улаз, „х“ за излазе).

У зависности од редног броја проблема **i** студент треба да реализује **један** од следећих алгоритама за стварање лавиринта:

0. Обилазак по дубини - „Depth-First Search“:

1. Кренути од чвора улаза – текући чвор је улазни
2. Означити чвор као „посећен“
3. Ако чвор има непосећених суседа:
 - i. На случајан начин одабрати једног од суседа
 - ii. Ставити тренутни чвор на стек
 - iii. Склонити зид између тренутног и изабраног чвора
 - iv. Изабрани чвор поставити за текући
 - v. Наставити на исти начин од текућег чвора
4. Ако нема непосећених суседа:
 - i. Скинути последњи чвор са стека и прогласити њега текућим
 - ii. Наставити на исти начин од текућег чвора

1. Примов алгоритам:

1. Почине се са мрежом зидова
2. Насумично изабрати чвор, означити га као део лавиринта и додати све зидове у скуп зидова лавиринта
3. Док год има зидова у листи:
 - i. Случајно изабрати зид из скупа
 - ii. Ако чвор са друге стране зида није у скупу чворова лавиринта:
 1. Од зида направити пролаз
 2. Означити чвор са друге стране као део лавиринта
 3. Додати зидове суседне ћелије у скуп зидова

2. Крускалов алгоритам:

1. Почине се од мреже зидова – сви зидови су додати у скуп зидова; сваки чвор се налази у свом засебном „скупу“
2. На случајан начин се бира зид:
 - i. Ако чворови које тај зид раздваја припадају различитим „скуповима“:
 1. Уклонити посматрани зид
 2. Спојити та два „скупа“ чворова

[25 поена] Решавање лавиринта

Потребно је реализовати и одговарајући алгоритам за решавање лавиринта. Решавање лавиринта подразумева налажење путање од улаза до излаза (било ког). 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: $(x0, y0) \rightarrow (x1, y1) \rightarrow (x2, y2) \rightarrow \dots \rightarrow (xK, yK)$, 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..

Исти алгоритам за решавање лавиринта треба применити и на следећи сценарио: Мали Перица се изгубио у лавиринту, уплашио се и не сме да се помери. Његов друг Јовица се понудио да му прискочи у помоћ, тако што ће доћи до њега да му прави друштво и отера страх. Као улазни параметри за решавање ове ситуације су чворови у којима се Перица и Јовица налазе, а излаз треба да буде истог формата као и приликом решавања целог лавиринта.

У зависности од редног броја проблема **i** студент треба да реализује **један** од следећих алгоритама за решавање лавиринта:

0. „Правило десне руке“:

1. Кренути од улазног чвора (или чвора где се Јовица налази)
2. Пратити десни зид, тј. ићи десно у односу на тренутни смер, кад год је то могуће
 - i. Ако није могуће, наставити право, односно лево, ако се не може ићи право
 - ii. Ако се дође у слепу улицу, треба се окренути и вратити до претходне раскрснице и одабрати други смер, и даље пратећи десни зид
3. Крај је када се дође до неког од чворова излаза

1. Обилазак по ширини:

1. Кренути од улазног чвора (или чвора где се Јовица налази) – текући чвор
2. Обележити чвор као „посећен“
 - i. Све суседне непосећене чворове ставити у ред
 - ii. Дохватити један чвор из реда
 - iii. Означити га као „посећен“
 - iv. Понављати поступак док год има непосећених чворова или док се не „посети“ чвор излаза

2. Флојд-Варшалов алгоритам

1. Кренути од улазног чвора или чвора где се Јовица налази – полазни чвор
2. Користећи Флојд-Варшалов алгоритам проверити да ли је неки од чворова излаза, односно чвор где се Перица налази, достижан из полазног чвора

[10 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.