

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1 2022-2023

- домаћи задатак 3 -

Опште напомене:

1. Домаћи задатак 3 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C.
2. 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**.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Унос података треба омогућити путем читања из датотеке.
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 успешно 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 матрице и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Одбрана трећег 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):
$$i = (R + G) \bmod 3 + 1$$
$$j = (R + G) \bmod 2 + 1$$
10. Име датотеке која се предаје мора бити **dz3p1.c**
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.

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

У зависности од редног броја **i** добијеног коришћењем формуле назначене у напоменама, потребно је користити једну од следећих меморијских репрезентација графа приликом решавања задатог 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 [35 поена]

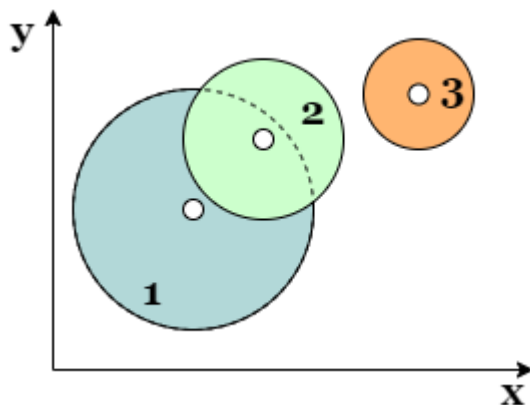
Написати програм на програмском језику C који илуструје рад са **усмереним** графовима. Програм треба да омогући следеће операције над графом:

- [5 поена] Креирање празне структуре података за граф задатих димензија (постоје чворови графа, али не постоје гране)
- [10 поена] Додавање чвора у граф и уклањање чвора из графа
- [10 поена] Додавање и уклањање гране између два чвора у графу
- [5 поена] Испис репрезентације графа
- [5 поена] 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 – Минско поље [65 поена]

Дат је стандардни Декартов координатни систем који представља минско поље. Мине се постављају на одређену позицију задавањем две реалне координате (x , y). Свака мина има свој радијус експлозије (реалан број) након детонације. Сматрати да је радијус правилан круг и да је интензитет експлозије константан на целој површини круга. Детонацијом једне мине, детонирају се и све мине које се налазе у њеном радијусу експлозије. На слици 1 приказане су три мине и одговарајући радијуси експлозије. На пример, детонацијом мине “1” се детонира и мина “2”. **Детонација мине може произвести даља детонирања.** Потребно је реализовати систем који проналази најефикаснију мину. Ефикасност представља укупан број детонираних мина (укључујући и мину која је прва детонирана). За разматрани пример, то је мина “1”, зато што се детонацијом мине “2” и “3” детонирају само оне саме (укупно по 1 мина), док се детонацијом мине “1” детонира и мина “2” (укупно 2 мине).



Слика 1. Координатни систем са три мине

Потребно је моделовати описани систем графом. Конкретан начин моделовања остављен је студенту и биће предмет оцењивања.

Имплементирати решење које ће омогућити:

- **[15 поена]** Учитавање података из улазне текстуалне датотеке и креирање графа према изабраном моделу.
- **[20 поена]** Проналажење ефикасности мине коју корисник задаје, уз испис детонираних мина.
- **[10 поена]** Проналажење мине максималне ефикасности.
- **[10 поена]** Експлозију ракете на задатом месту (корисник задаје координате (x, y) и радијус експлозије ракете). Сматрати да је ракета експлодира на исти начин као и мина. Потребно је исписати детониране мине.
- **[10 поена]** Проналажење укупне површине која је покривена експлозијом мине максималне ефикасности. Обратити пажњу на то да ће површине кругова бити преклопљене.

У зависности од редног броја **j** добијеног коришћењем формуле назначене у напоменама, потребно је користити једну од следећих обилазака за проналажење ефикасности мина:

1. Алгоритам за обилазак графа по ширини
2. Алгоритам за обилазак графа по дубини

Улазна текстуална датотека у првом реду садржи број мина n , и у наредних n редова торку (x, y, r) , реални бројеви раздвојени размацама, која представља координате и радијус експлозије одговарајуће мине. Студентима су на располагању графови за тестирање у датотекама *graf1.txt* и *graf2.txt*, а сваки студент треба да осмисли један додатан пример графа за тестирање са минимално 10 мина.

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.**