

**ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО**  
**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1**  
**2023-2024**

- трећи домаћи задатак -

**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 и/или 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 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):
$$i = (R + G) \bmod 4 + 1$$
$$j = (R + G) \bmod 3 + 1$$
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. Уланчану репрезентацију коришћењем инверзних листа суседности
4. 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 [40 поена]

Написати програм на програмском језику C који илуструје рад са **нетежинским неусмереним** графовима. Програм треба да омогући стварање графа читавањем са стандардног улаза, основне операције попут додавања нових и брисања постојећих чворова и грана, испис репрезентације графа, као и брисање целог графа. Приликом читавања графа, корисник наводи број чворова и грана, а затим за сваки чвор уноси редне бројеве чворова-суседа. Сматрати да су чворови нумерисани од нуле. 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)
- [15 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 – Анализа социјалних мрежа [60 поена]

Теорија мрежа (енг. *network science*) је примењена грана меморије графова која се бави проучавањем усмерених и неусмерених графова којима се моделују односи између дискретних објеката (ентитета). Анализа социјалних мрежа представља скуп техника произашлих из примене теорије мрежа на ентитете (чворове) који представљају људе и њихове активности (ране). Комбинује знања из теорије графова, статистичке физике, истраживања података, визуализације информација и социологије. Социјалне мреже поседују нетривијалне тополошке одлике које се квантитативно могу описати различитим мрежним метрикама.

Потребно је допунити претходно реализовани програм, тако да укључи и одређене сложеније графовске операције и метрике за анализу социјалних мрежа. Описи појединих сложенијих графовских операција и метрика су дати у наставку текста.

**Густина графа** се рачуна као количник укупног броја грана у графу и максималног могућег броја грана у посматраном графу. Граф има максимални број грана ако између свака два чвора постоји грана.

**Достижност чвора** (или повезаност) за неусмерене графове показује да ли постоји путања између посматраног чвора и преосталих чворова графа. Приликом налажења достижности треба исписати све чворове који су достижни из задатог чвора, као и путању којом се до њих дошло.

**Фрагментација графа** представља количник броја парова чворова који су међусобно достижни у посматраном графу и укупног броја парова чворова.

**Геодезијска дистанца два чвора** представља најкраће растојање између њих у датом графу. За задати пар чворова треба исписати све чворове који се на тој путањи налазе, као и вредност геодезијске дистанце. Уколико чворови нису достижни (повезани), геодезијска дистанца не постоји, што треба и исписати.

**Дијаметар графа** је једнак највећој геодезијској дистанци у графу.

**Централност чвора по степену** за неусмерени граф представља број свих грана које су инцидентне посматраном чвору:

$$C_D(p_i) = \sum_{k=1}^N A(p_i, p_k)$$

где  $A(p_i, p_k)$  означава да постоји грана између чворова  $p_i$  и  $p_k$ , при чему је  $p_i$  посматрани чвор, а  $N$  представља број чворова у графу. Уколико постоји грана  $A(p_i, p_i)$ , рачуна се као да су то две гране.

**Централност чвора по блискости** представља количник укупног броја чворова умањеним за један и суме свих геодезијских дистанци од посматраног чвора ка свим осталим чворовима са којима је повезан:

$$C_C(p_i) = \frac{N - 1}{\sum_{k=1}^N d(p_i, p_k)}$$

где  $N$  представља број чворова, а  $d(p_i, p_k)$  представља најкраће растојање између посматраног чвора  $p_i$  и чвора  $p_k$  са којим је посматрани повезан.

**Релациона централност чвора** представља број најкраћих путања између свих осталих чворова на којима се посматрани чвор налази:

$$C_B(p_i) = \sum_{j=1}^N \sum_{k=1}^{j-1} g_{jk}(p_i)$$

где је  $g_{jk}$  најкраћи пут између чворова  $p_j$  и  $p_k$ , а  $g_{jk}(p_i)$  означава да ли је на том путу укључен и чвор  $p_i$ .

Зависно од редног броја проблема **j** који се решава, треба реализовати следеће комбинације операција над графом:

1. Геодезијска дистанца чворова, дијаметар графа, релациона централност чвора
2. Густина графа, геодезијска дистанца чворова, централност чвора по блискости
3. Достижност чвора, фрагментација графа, централност чвора по степену

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.