

**ОДСЕК ЗА РАЧУНАРСКУ ТЕХНИКУ И ИНФОРМАТИКУ**  
**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА**  
**2024-2025**  
**- домаћи задатак -**

**Опште напомене:**

1. Домаћи задатак састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C++.
2. Право да раде домаћи задатак имају сви студенти који прате предмет. Предаја домаћих задатака ће бити могућа до **недеље, 8.12.2024.** коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да се пријаве на курс пре термина одбране домаћег задатка.** Пријава на курс ће бити прихваћена и важећа само уколико је студент регистрован на систем путем свог налога електронске поште на серверу *mail.student.etf.bg.ac.rs*. Име архиве која се предаје мора бити ***piggbbbb.zip***. Погрешно назван и предат домаћи задатак повлачи одузимање 25 поена. Прецизније информације везане за пријаву и предају домаћег ће бити накнадно објављене.
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 било путем читања са стандардног улаза, било путем читања из текстуалне 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 и неће бити оцењена са максималним бројем поена.**
7. **Није дозвољено коришћење готових структура података из *std* и осталих библиотека.**
8. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити које претпоставке и која ограничења су усвојена (на пример, максимална димензија низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
9. Одбрана домаћег задатка ће се обавити у **понедељак, 9.12.2024.** према распореду који ће накнадно бити објављен на сајту предмета.
10. Формула за редни број проблема ***i*** који треба решавати је следећа (***R*** – редни број индекса, ***G*** – последње две цифре године уписа):

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

11. 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. Током израде решења није дозвољена употреба алата вештачке интелигенције заснованих на великим језичким моделима (*ChatGPT*, *Github Copilot* и сл.).

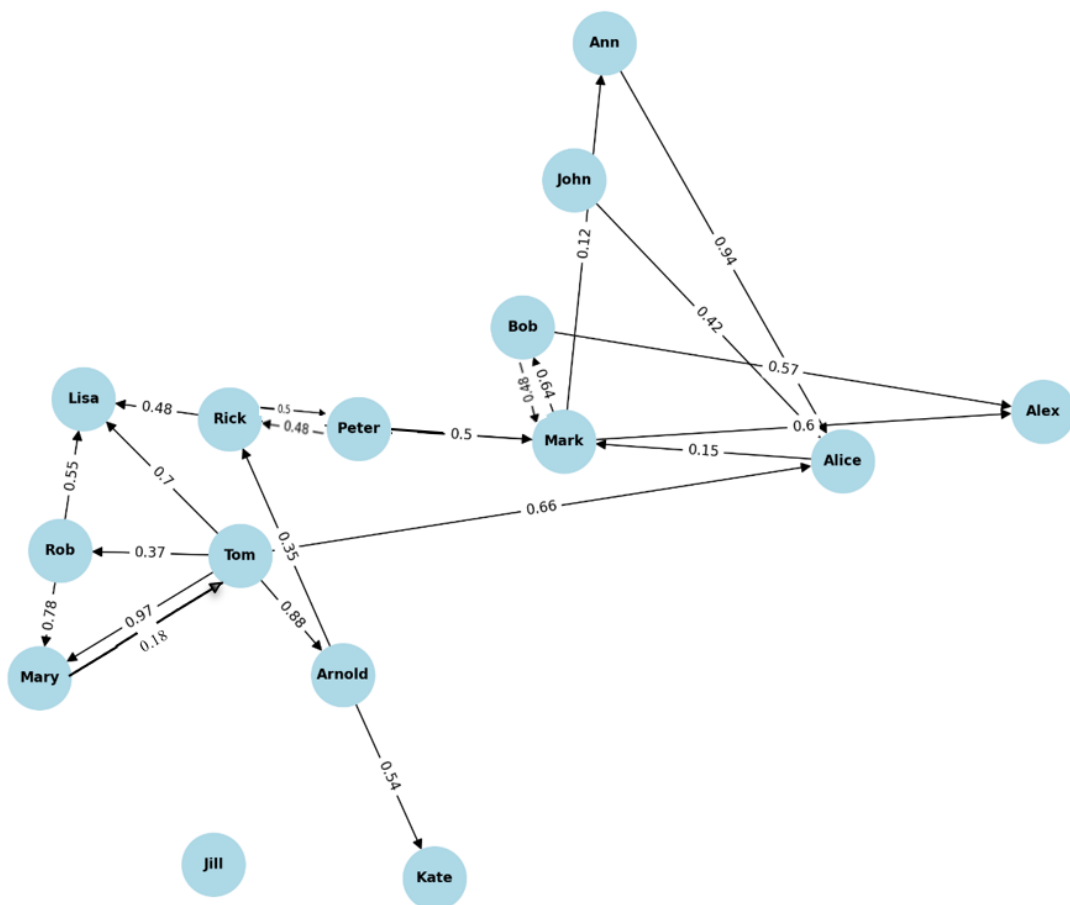
## Моделовање друштвене мреже коришћењем графова

Моделовање друштвених односа и интеракција у циљу анализе понашања у мрежама предмет је интензивног истраживања. Један од начина да се друштвена мрежа моделује јесте усмереним тежинским графом, где се корисници представљају чворовима графа, а везе између корисника гранама између тих чворова. Односи у друштвеној мрежи могу представљати различите типове интеракција, као што су пријатељства, односно праћење (следбеништво), а може се узети у обзир и учесталост комуникације.

Тежина придружена грани моделује интензитет или важност односа, тако да веће тежине грана представљају јачу или учесталију интеракцију између корисника које та грана повезује. Тежине се најчешће изражавају реалним бројевима у опсегу од 0 до 1, где је 0 минимална, а 1 максимална јачина везе.

Нека се посматра једна друштвена мрежа заснована на праћењу. Најближи корисник одређеном кориснику је онај који има највећу тежину везе са њим. На пример, ако постоји усмерена грана од чвора А до чвора Б са тежином 0.75, то значи да ће објава коју је објавио корисник Б са вероватноћом од 75% изаћи на почетној страници корисника А. Уколико се кориснику А свиђа објава коју је корисник Б поделио, вероватноћа да кориснику А убудуће излазе објаве корисника Б повећава се за 0.1.

Када се посматра повезаност корисника А и Ц преко корисника Б, тежине интеракција дуж тог пута се множе како би се израчунала индиректна снага повезаности. Овакав приступ омогућава дубљу анализу мрежне структуре, односа између корисника и начина ширења информација.



Слика 1. Пример једне друштвене мреже моделоване графом

## Формирање и манипулација графом [55 поена]

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“.

Написати програм на програмском језику C++ који илуструје рад са усмереним тежинским графовима. Програм треба да омогући следеће операције над графом који представља једну лексичку базу знања:

- [15 поена] Учитавање скупа корисника и релација између њих из текстуалног фајла задатог формата и формирање графа.
- [10 поена] Додавање чвора у граф и уклањање чвора из графа.
- [10 поена] Додавање и уклањање гране између два чвора у графу.
- [10 поена] Испис репрезентације графа.
- [10 поена] Брисање графа из меморије.

Формат текстуалног фајла из којег се читају подаци је следећи. У првом реду текстуалног фајла се налази број чворова графа ( $n$ ), у другом реду број грана ( $e$ ), у трећем реду се налази  $n$  стрингова који су идентификатори чворова, а наредних  $e$  редова чувају информације о јачини везе између пара корисника. Информације о јачини везе су у формату *особа1 особа2 јачина\_везе*, где су *особа1* и *особа2* идентификатори чворова, а *јачина\_везе* јачина везе од корисника са идентификатором *особа1* до корисника са идентификатором *особа2*. Сматрати да идентификатор корисника не садржи више од 20 карактера, као и да су редови сортирани лексикографски.

## Интеракција са друштвеном мрежом [45 поена]

У циљу налажења повезаности одређених корисника, симулације рада и анализе друштвене мреже, потребно је омогућити следеће операције над друштвеном мрежом:

- [5 поена] Акцију „лајковања“ објаве корисника од стране другог корисника којом један корисник означава да му се свиђа објава другог корисника.
- [10 поена] Налажење највеће компоненте у графу која представља подкуп корисника који су сви међусобно повезани.
- [10 поена] Испис свих корисника на највероватнијем путу од једног задатог корисника до другог задатог корисника, у формату:  
*корисник1 - (вероватноћа) -> корисник2 - (вероватноћа) -> корисник3 -> ...*
- [20 поена] Ефикасно одређивање  $k$ -те особе ( $k$  уноси корисник), са највећим утицајем на друштвеној мрежи. Особа има највећи утицај када њена објава са највећом вероватноћом у односу на објаве осталих корисника у мрежи може доћи до свих корисника у мрежи.

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.

## Напомене

По потреби реализовати и додатне методе, где је то примерено.

За тестирање програма се могу користити датотеке са табелама различитих величина, које се налази у оквиру посебне архиве.

Рад са датотекама у језику C++ захтева увожење заглавља `fstream` (именски простор `std`). За читање података користи се класа `ifstream`. Након отварања датотеке, читање се врши на исти начин као и са стандардног улаза. Кратак преглед најбитнијих метода и пријатељских функција ове класе је дат у наставку.

<pre>void open(     const char * _Filename,     ios_base::openmode _Mode = ios_base::in,     int _Prot = (int)ios_base::_Openprot );</pre>	Отвара датотеку задатог имена за читање. <code>ifstream dat;</code> <code>dat.open("datoteka.txt");</code>
<pre>void close();</pre>	Затвара датотеку.
<pre>bool is_open();</pre>	Утврђује да ли је датотека отворена.
<pre>operator&gt;&gt;</pre>	Преклопљен оператор за просте типове података.
<pre>ifstream dat; dat.open("datoteka.dat"); if( ! dat.is_open() ) greska(); char niz[20]; dat &gt;&gt; niz; dat.close();</pre>	Пример отварања датотеке, провере да ли је отварање успешно, читање једног знаковног низа из датотеке и затварања датотеке.