

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1
2021-2022
- домаћи задатак 3 -

Опште напомене:

1. Домаћи задатак 3 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C.
2. Приликом предаје домаћег задатка 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 је студент регистрован на систем путем свог налога електронске поште на серверу mail.student.etf.bg.ac.rs.
3. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Решење треба да буде отпорно на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
5. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. Примена рекурзије се неће признати као решење проблема које може освојити 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 домаћег задатка ће се обавити у **понедељак, 30.05.2022.** према распореду који ће накнадно бити објављен на сајту предмета.
8. За решавање задатака који имају више комбинација користити следеће формуле.
(**R** – редни број индекса, **G** – последње две цифре године уписа):

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

$$j = (R + G) \bmod 2 + 1$$

9. Име датотеке која се предаје мора бити **dz3p1.c**
10. Предаја домаћих ће бити омогућена преко Moodle система до **понедељка, 30.05.2022. у 10:00.** Детаљније информације ће бити благовремено објављене.
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.

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 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 [40 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

[5 поена] 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 – Мултилатерална компензација [60 поена]

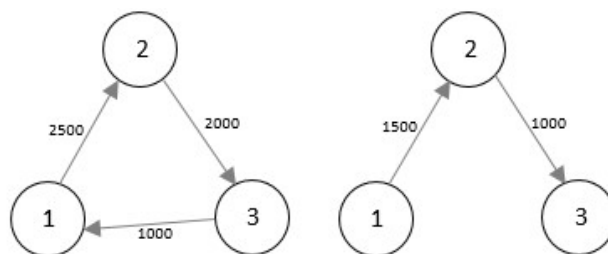
Мултилатерална компензација представља начин пребијања дуговања (потраживања) између више пословних субјеката. Извршавање компензације подразумева проналажење највећег могућег износа који се може компензовати, односно пребити. Потраживања међу пословним субјектима се могу моделовати усмереним тежинским графом. Грана од субјекта X ка субјекту Y тежине T означава постојање дуга износу од T евра (или друге новчане јединице) пословног субјекта X субјекту Y .

На пример, нека су пре компензације дуговања следећа (слика 1, лево):

- Пословни субјект 1 има дуговање према пословном субјекту 2 у износу од 2500 евра,
- Пословни субјект 2 има дуговање према пословном субјекту 3 у износу од 2000 евра и
- Пословни субјект 3 има дуговање према пословном субјекту 1 у износу од 1000 евра.

Дакле, максимално је могуће компензовати износ од 1000 евра. Након компензације, стање је следеће (слика 1, десно):

- Пословни субјект 1 има дуговање према пословном субјекту 2 у износу од 1500 евра,
- Пословни субјект 2 има дуговање према пословном субјекту 3 у износу од 1000 евра и
- Пословни субјект 3 више нема дуговања према пословном субјекту 1.



Слика 1. Пример графа који илуструје потраживања између пословних субјеката:
(лево) пре компензације, (десно) после компензације

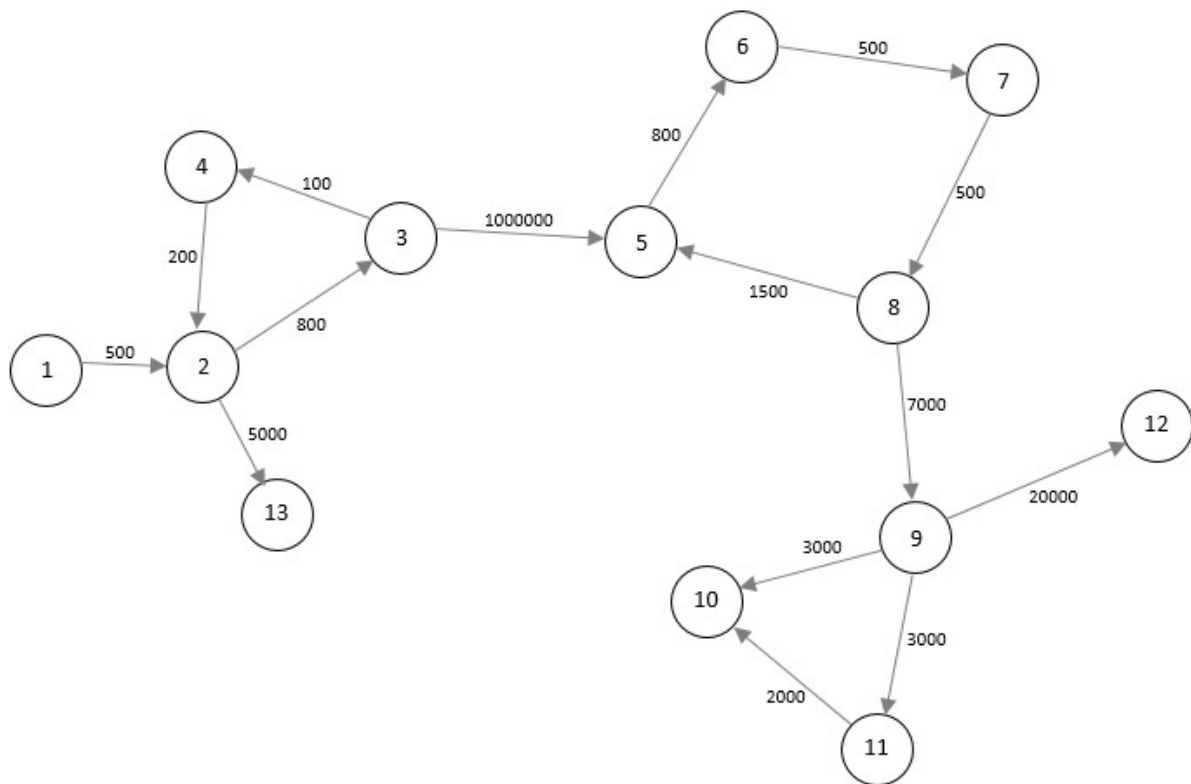
Имплементирати решење које ће омогућити:

- **[10 поена]** Учитавање графа који представља дуговања компанија из улазне текстуалне датотеке
- **[20 поена]** Проналажење и исписивање циклуса у графу
- **[10 поена]** Одређивање највећег могућег новчаног износа који је могуће компензовати за сваки пронађени циклус
- **[20 поена]** Спровођење компензације и формирање новог графа дуговања.

У зависности од редног броја j добијеног коришћењем формуле назначене у напоменама, потребно је користити једну од следећих техника за проналажење циклуса у усмереном графу:

1. Алгоритам за обилазак графа по дубини
2. *Warshall*-ов алгоритам

Сматрати да се у оквиру улазне текстуалне датотеке граф репрезентује матрицом суседности по следећем формату: у првом реду датотеке се налази број n који представља број чворова у графу, а затим следи n редова који садрже саму матрицу суседности. Студентима су на располагању графови за тестирање у датотекама *graf2.txt* и *graf3.txt*, а сваки студент треба да осмисли један додатан пример графа за тестирање са минимално 10 чворова и 3 циклуса. На слици 2 је приказан граф дуговања који се налази у датотеци *graf2.txt*.



Слика 2. Пример графа дуговања