

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1
2016-2017
- 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 два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику С.
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 пруже 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, 29.05.2017.** према распореду који ће бити објављен на сајту предмета.
8. Формула за редни број репрезентације графа **i** коју треба користити приликом решавања **првог** и **другог** задатка је следећа: (R – редни број индекса, G – последње две цифре године уписа):
$$i = (R + G) \bmod 3 + 1$$
9. Имена датотека које се предају морају бити **dz3p1.c** и **dz3p2.c**
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

19.05.2017. 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

У зависности од редног броја 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 (60 поена)

Написати програм на програмском језику C који илуструје рад са тежинским неусмереним графовима. Програм треба да омогући стварање графа учитавањем са стандардног улаза, неке основне операције попут додавања нових и брисања постојећих чворова и грана, као и брисање целог графа. Приликом учитавања графа, корисник наводи број чворова и грана, а затим за сваки чвор уноси редне бројеве чворова-суседа и тежину гране која их спаја. Сматрати да су чворови нумерисани од нуле и да су тежине грана ненегативни цели бројеви. 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, тј:

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

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, тј:

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

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, тј:

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

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 i 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.

Задатак 2 – Путарина (40 поена)

Путник жели да стигне од почетног места до одредишта и поседује извесну своту новца M . Да би пронашао најбољи пут, посматра мапу свих места у том региону. За свака два места уцртана на карти између којих постоји пут дата је и њихова раздаљина. При уласку у неко место на изабраној путањи, путник мора да плати одређену суму новца. Он жели на циљ да стигне најкраћим путем, али мора да води рачуна и о цени коју на том путу треба да плати. Уколико нема довољно новца да уђе у неко место, путник остаје заробљен испред, те због тога он мора да пронађе најкраћу путању за коју има довољно новца. Уколико постоји више најкраћих путања којима путник може стићи до циља, он бира најјефтинију. При израчунавању цене целокупне путање не узима се у обзир цена почетног чвора. Потребно је исписати најкраћу путању за коју корисник има новца, заједно са новцем који му на крају преостаје, а уколико таква путања не постоји, треба исписати одговарајућу поруку. Осим података потребних за формирање графа, као улазне параметре потребно је навести и колико новца путник има на располагању, као и почетни и крајњи чвор.

У наставку је дат пример једне мапе. Претпоставимо да путник жели да из места A стигне у место E , са 60 новчића на располагању. За конкретан граф, могао би да изабере најкраћу путању ($A-B-E$), међутим, уколико има 40 новчића могао би да изабере нешто дужу путању $A-D-C-E$. Са 25 новчића, доступне су му путање $A-C-E$ за 20 и $A-D-E$ за 25. Обе су исте дужине, па према алгоритму треба да изабере јефтинију. Ако креће на пут са 17 новчића у цепоу, он ни не може да стигне до чвора E .

