

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1 2017-2018

- 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 једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C.
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 пружи 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, 03.06.2019.** и **уторак, 04.06.2019.** према распореду који ће бити објављен на сајту предмета.
8. Формула за редни број репрезентације **i** коју треба користити приликом решавања задатка је следећа: (R – редни број индекса, G – последње две цифре године уписа):
$$i = (R + G) \bmod 3$$
9. Формула за редни број алгорита **j** који треба користити приликом решавања задатка је следећа: (R – редни број индекса, G – последње две цифре године уписа):
$$j = (R + G) \bmod 2$$
10. Име датотеке која се предају мора бити **dz3p1.c**
11. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

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

Граф се може репрезентовати коришћењем матрица суседности или листа суседности. У оквиру овог задатка, студенти треба да користе репрезентацију графа у зависности од редног броја проблема **i**:

0. Матрица суседности
1. Листе суседности
2. Линеаризоване листе суседности

Више информација о меморијским репрезентацијама графа се може пронаћи у материјалима са предавања и вежби, као и у књизи проф. Мила Томашевића „Алгоритми и структуре података“.

Задатак - Bloxorz (100 поена)

Написати програм на програмском језику C који симулира једну варијанту игре Bloxorz (www.bloxorz.org.uk/). Игра се заснива на кретању блока димензија 1x1x2 по терену комплексног облика. Терен за игру се састоји из плоча по којима се блок креће котрљањем помоћу четири команде(горе, доле, лево, десно). Циљ игре је довести блок од стартне до циљне позиције. Игра се може завршити на два начина:

- победом – када блок доспе у циљну позицију тако да је на циљној плочи постављен усправно
- поразом – када се блок помери изван граница терена.

Мапа терена се може моделирати коришћењем неусмереног графа. Моделира се чворовима који представљају плоче – из сваког чвора се може прећи у суседни леви, десни, горњи или доњи чвор, под условом да између њих постоји грана, односно ако су плоче једна до друге.

[50 поена] Основне операције над графом

Програм треба да садржи и користи помоћне функције за стварање графа, додавање чворова, додавање грана, брисање чворова, брисање грана и брисање целог графа. Стварање графа реализовати читањем мапе терена из фајла. Пример записа мапе терена дат је на следећој слици:

```
-----  
-ooo-----  
-oSoooo-----  
-ooooooooo----  
--ooooooooo---  
-----ooTo--  
-----ooo----  
-----
```

Мапа терена записана је као матрица следећих симбола:

- симбол „o“ који представља плочу
- симбол „-“ који означава да на датој позицији није постављена плоча
- симбол „S“ који означава стартну позицију (блок је на стартној плочи иницијално постављен усправно)
- симбол „T“ који означава крајњу позицију

Након стварања графа, омогућити додавање плоче на задату позицију на ивици терена, као и уклањање задате плоче са ивице терена. Омогућити испис модификованих мапа на стандардни излаз или у текстуалну датотеку, по истом формату као што је усвојено за улазни фајл. У складу са потребама задатка, усвојити и имплементирати структуру података за репрезентацију графа над којом ће бити спровођене задате операције.

[40 поена] Имплементација игре

Програм треба да омогући одигравање потеза, односно померање блока у зависности од потеза који корисник зада (горе, доле, лево, десно), уз приказ стања игре након одигравања потеза. Корисник задаје потез уношењем бројева (1-4) или знакова (ASDW). Приказ стања игре подразумева исцртавање мапе терена на којој се види где се блок налази. Уколико се након одигравања блок доведе у циљну позицију, тако да је на циљној плочи постављен усправно, игра се завршава победом. Уколико се након одигравања блок избаци ван терена, игра се завршава поразом.

Потребно је реализовати и одговарајући алгоритам за решавање игре. Решавање подразумева налажење секвенце потеза који доводе блок из почетне у крајњу позицију. Решење не мора да постоји, и у том случају треба исписати одговарајућу поруку на стандардни излаз. Ако постоји решење, треба исписати низ потеза које треба одиграти.

У зависности од редног броја проблема **ј** студент треба да реализује **један** од следећих алгоритама за решавање игре:

0. Обилазак графа по дубини
1. Обилазак графа по ширини

[10 поена] Комуникација са корисником

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.