

**ОДСЕК ЗА РАЧУНАРСКУ ТЕХНИКУ И ИНФОРМАТИКУ**  
**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА**  
**2020-2021**  
**- домаћи задатак -**

**Опште напомене:**

1. Домаћи задатак састоји се од два програмска проблема. Студенти проблем решавају **самостално**, на програмском језику C++. **Дозвољено је коришћење готових структура података из STL и сличних библиотека.**
2. Право да раде домаћи задатак имају сви студенти који прате предмет. Предаја домаћих задатака ће бити могућа до **уторка, 15.12.2020.** Прецизније информације везане за пријаву и предају домаћег ће бити накнадно објављене.
3. Домаћи задатак замењује други део испита, односно задатак на испиту који се пише у конкретном језику. Поени са домаћег задатка ће стога бити скалирани у одговарајућем односу. Одбраном домаћег задатка студент се одриче права да ради други део испита. Студент на испиту треба да назначи да ли је радио домаћи задатак како би му били признати поени.
4. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
5. Унос података треба омогућити путем читања са стандардног улаза.
6. Решења треба да буду отпорна на грешке и треба да кориснику пруже јасно обавештење у случају детекције грешке.
7. 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.**
8. 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 низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
9. Одбрана домаћег задатка ће се обавити у **четвртак, 17.12.2020.**
10. Формула за редни број комбинације проблема **i** и **j** који треба решавати је следећа (R – редни број индекса, G – последње две цифре године уписа):
$$i = (R + G) \bmod 3 + 1$$
$$j = (R + G) \bmod 2 + 1$$
11. Имена датотеке које се предаје мора бити **dzp1.cpp** и **dzp2.cpp**
12. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

## Графови [100 поена]

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 [60 поена]

Написати програм на програмском језику C++ који илуструје рад са неусмерним графовима. Програм треба да омогући следеће операције над графом:

- [5] Креирање празне структуре података за граф задатих димензија (постоје чворови графа, али не постоје гране)
- [15] Додавање чвора у граф и уклањање чвора из графа
- [15] Додавање и уклањање гране између два чвора у графу
- [10] Испис репрезентације графа
- [10] 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 - Проблем изградње путне инфраструктуре [40 поена]

У главном граду једне мале државе честе су велике гужве у саобраћају. Грађани су приметили да губе много времена путујући градским превозом и већ дуже време захтевају од власти да направе метро. Након много година одлагања, пројекат изградње метроа је почео да се разматра. Од инжењера је затражено да се израчуна минимална цена која је потребна како би метро повезао кључне тачке у граду. Такође, у граду постоји проблем са брзином одзива хитних служби (ватрогасци, хитна помоћ и сл.) због неадекватне путне инфраструктуре. Коришћењем претходно реализованих функција за рад са графом, потребно је формирати граф и омогућити решавање наведених проблема према траженим захтевима.

### [20 поена] Стварање и испис метроа

Корисник учитава податке о кључним локацијама у граду са стандардног улаза. Најпре се уноси број кључних локација у граду, а затим је потребно учитати између којих све локација је **могућа** изградња метроа као и цена уколико се изгради. Након учитавања података, потребно је исписати минималну цену потребну за изградњу метро железнице тако да повезује све учитане тачке.

У зависности од редног броја  $j$  добијеног коришћењем формуле назначене у напоменама, потребно је користити један од следећих алгоритама:

1. Примов алгоритам (насумично изабрати почетни чвор).
2. Крускалов алгоритам

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

### [20 поена] Проналажење најкраће руте за ватрогасна кола

Након израчунавања оптималне цене за метро, инжењери су анализирали оптимизацију путања ватрогасних кола до кључних локација у граду. Познато је да се сва ватрогасна кола паркирају на само једној од локација у граду ( $V$ ). Инжењери желе да израчунају најкраће растојање до сваке локације (посебно) од тачке  $V$ .

У овом граду друмски саобраћај (чиме иду ватрогасна кола) је делимично изграђен. У складу са тим, постоје парови локација између којих не постоји друмски саобраћај, а могуће га је изградити.

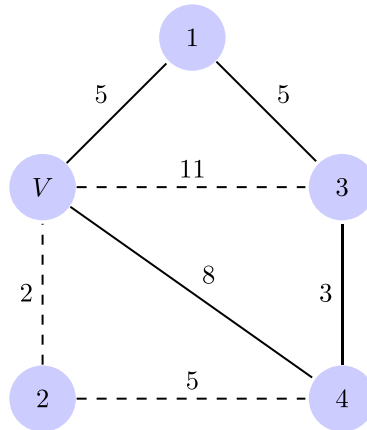
Корисник учитава податке о кључним локацијама у граду за друмски саобраћај са стандардног улаза. Потребно је учитати:

1. индекс локације  $V$
2. локације између којих већ **постоји** изграђен пут и његову дужину.
3. локације између којих **је могуће** изградити пут и његову дужину.

**Напомена:** Кључне локације су еквивалентне као у претходном задатку за изградњу метроа, али њихове везе нису, јер се ради о другом типу саобраћаја.

За сваку **локацију посебно**, потребно је пронаћи најкраћи пут од тачке  $V$ . Уколико постоји више решења, исписати сва, сортирано по броју неизграђених путева.

## Пример:



Пуном линијом је представљен **изграђен** друмски пут, испрекиданом линијом је представљен **неизграђен** друмски пут.

За локацију 1 најкраћи пут је директан пут  $V - 1$  (дужина 5).

За локацију 2 најкраћи пут је директан пут који се мора изградити  $V - 2$  (дужина 2)

За локацију 3 најкраћи пут је дужине 10. Постоје два таква пута и потребно их је исписати у следећем редоследу:  $V - 1 - 3$ ,  $V - 2 - 4 - 3$ . Пут  $V - 1 - 3$  садржи 0 неизграђених путева, пут  $V - 2 - 4 - 3$  садржи 2 неизграђена пута.

За локацију 4 најкраћи пут је:  $V - 2 - 4$  (дужина 7).

### 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.