

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2

2025-2026

- први домаћи задатак -

Опште напомене:

1. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику С или С++.
2. 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.** Пријава на курс ће бити прихваћена и важећа само уколико се студенат региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
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 датотеке.
5. Решења треба да буду отпорна на грешке и треба да кориснику пруже 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 које може освојити максималан број поена.**
7. 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.
8. Одбрана првог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
9. За решавање задатака који имају више комбинација користити следеће формуле. (**R** – редни број индекса, **G** – последње две цифре године уписа):

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

$$j = (R + G) \bmod 4$$

10. Имена датотека која се предају морају бити **asp2dz1p1.c(pp)** и **asp2dz1p2.c(pp)**. Погрешно назван и предат домаћи задатак се кажњава са одузетих 25 поена.
11. Предаја домаћих ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене.
12. 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.).

Задатак 1 – мониторинг температурних промена [40 поена]

Метеоролошка станица прати температуру ваздуха 365 дана у години. Сваког дана у тачно подне (12:00) забележи се температура у целим степенима Целзијуса. На крају године, подаци се уређују и чувају као **уређени низ температура** сортиран неопадајуће.

Прецизна структура записа о температурном мерењу се оставља студенту на имплементацију.

Потребно је имплементирати следеће функционалности [40 поена]:

- [10 поена] **Учињавање температурних зајиса**

Корисник задаје број мерења (до 365) и температуре у неопадајућем редоследу.

- [10 поена] **Генерисање температурних зајиса**

Генерисање уређеног низа температура у задатом опсегу. Корисник задаје број мерења и температурни опсег. Програм генерише температуре у задатом опсегу на псеудослукајан начин, а затим их сортира неопадајуће. Дозвољено је понављање температура.

- [10 поена] **Пређирање температурног рекорда**

Проналажење да ли је одређена температура забележена у току године и на којим позицијама у уређеном низу се налази. Уколико температура постоји на више позиција (забележена више дана), пронаћи **прву** и **последњу** позицију на којој се јавља.

У зависности од редног броја *i* проблема који се решава, реализовати претрагу применом једне од следећих техника:

0. **Бинарна претрага** - класична бинарна претрага са модификацијом за проналажење прве и последње појаве циљне температуре у уређеном низу.
1. **Тернарна претрага** - модификована бинарна претрага (дели низ на три дела уместо два) са проналажењем прве и последње појаве циљне температуре.
2. **Интерполационна претрага** - претрага која користи процену позиције циљне вредности на основу њене релативне величине у опсегу.

- [10 поена] **Пређирање у температурном опсегу**

Проналажење свих температура које су у задатом опсегу [*temp_min*, *temp_max*]. На пример, пронаћи све дане када је температура била између 20°C и 30°C. Користити технику претраге из претходне функционалности.

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.

Задатак 2 – имплементација стабла бинарног претраживања за програм лојалности хотела [60 поена]

Хотел "Балкан" има програм лојалности за честе госте. Сваки гост сакупља поене за сваки боравак у хотелу. Систем чува информације о гостима у оквиру стабла бинарног претраживања. Сваки запис о госту садржи барем: *број поена лојалности (кључ), име и презиме госта*. Сматрати да не постоје два госта са истим именом, презименом и бројем поена. Прецизна структура чвора стабла се оставља студенту на имплементацију. У оквиру чвора стабла постоји показивач на лево и десно подстабло, као и показивач на оца.

Потребно је подржати следеће операције:

- **[15 поена]** Формирање стабла на основу листе гостију учитане са стандардног улаза.
- **[5 поена]** Претрага госта у стаблу на основу броја освојених поена лојалности.
- **[20 поена]** Увећавање поена госта за одређен број поена. Водити рачуна да структура бинарног стабла остане валидна.
- **[5 поена]** Форматиран испис садржаја стабла.
- **[5 поена]** Брисање стабла из меморије.

У зависности од редног броја j проблема који се решава, имплементирати једну од следећих додатних функционалности **[10 поена]**:

0. Проналажење свих гостију са бројем поена у задатом опсегу $[min_points, max_points]$.
1. Проналажење K -тих госта по броју поена.
2. Одређивање \bar{K} -тих госта са задатим бројем поена у укупној листи гостију сортираној по поенима.
3. За одређеног госта, пронаћи госта са најближим већим бројем 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.