

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2**2023-2024****- први домаћи задатак -****Опште напомене:**

1. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику C или C++.
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 студент региструје 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. Унос података треба омогућити путем читања са стандардног улаза.
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):

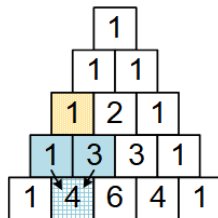
$$i = (R + G) \bmod 3$$

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

10. Имена датотека која се предају морају бити **asp2dz1p1.c(pp)** и **asp2dz1p2.c(pp)**
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.

Задатак 1 – претрага линеарних структура података [45 поена]

Написати програм за поређење перформанси различитих техника претраге задатог реда Лозанићевог троугла (https://en.wikipedia.org/wiki/Lozani%C4%87%27s_triangle). Лозанићев троугао се формира слично Паскаловом троуглу (https://en.wikipedia.org/wiki/Pascal%27s_triangle). Паскалов троугао је бесконачан низ природних бројева у облику троугаоне шеме, где сваки број у једном реду представља збир два броја изнад њега, а крајњи бројеви шеме су увек јединице.

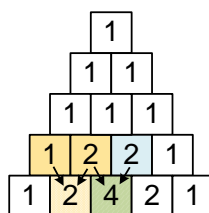


Слика 1. Паскалов троугао реда 4

На слици 1 је приказан Паскалов троугао, закључно са четвртим редом (бројање креће од нуле). За израчунавање k -тог елемента у реду n може се применити формула:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Лозанићев троугао такође по ивицама има јединице и већина бројева се формира као у Паскаловом, уз изузетак бројева на непарним позицијама у парним редовима (бројање креће од 0) – сваки број на непарној позицији p у парном реду r се рачуна тако што се од суме два броја изнад (као у Паскаловом) одузме број који се налази у $\left(\frac{r}{2} - 1\right)$ -ом реду на позицији $\frac{p-1}{2}$ из Паскаловог троугла.



Слика 2. Лозанићев троугао реда 4

На слици 2 је приказан Лозанићев троугао, закључно са четвртим редом. На слици 1 је означен број Паскаловог троугла који утиче на формирање одговарајућег означеног броја на слици 2 (позиција 1 у реду 4).

Ако је $T(n, k)$ број Лозанићевог троугла у реду n на позицији k , он се рачуна као

$$T(n, k) = \begin{cases} T(n-1, k-1) + T(n-1, k) - \binom{\frac{n}{2}-1}{\frac{k-1}{2}}, & \text{за } n \bmod 2 = 0 \text{ и } k \bmod 2 = 1 \\ T(n-1, k-1) + T(n-1, k), & \text{иначе} \end{cases}$$

тј. $T(n, k) = \frac{1}{2} \left[\binom{n}{k} + \binom{n \bmod 2}{k \bmod 2} \binom{\lfloor \frac{n}{2} \rfloor}{\lfloor \frac{k}{2} \rfloor} \right]$ (https://oeis.org/wiki/Lozani%C4%87%27s_triangle).

Критеријум за поређење перформанси је просечан број приступа приликом успешне претраге. На пример, претрага четвртог реда за кључ 2 ће бити успешна, док ће претрага истог реда за кључ 3 бити неуспешна.

У зависности од редног броја проблема i који се решава, реализовати следеће технике претраживања [20 поена]:

0. Бинарно и интерполационо претраживање
1. Тернарно и интерполационо претраживање
2. Бинарно и тернарно претраживање

Демонстрацију рада програма остварити кроз једноставан интерактивни мени који ће омогућити комуникацију са корисником. Кориснику треба омогућити следеће операције:

- [10 поена] Унос редног броја реда Лозанићевог троугла за претраживање и генерисање секвенце кључева за претраживање.
- [10 поена] Покретање и испис корака симулације, тј. претраге.
- [5 поена] Испис резултата и поређење перформанси на стандардном излазу.

Поступак се понавља све док корисник у менију не изабере опцију за прекид програма.

Задатак 2 – имплементација скупова коришћењем AVL стабла [55 поена]

Написати програм за рад са скуповима целих бројева представљених AVL стаблом. Потребно је подржати следеће операције за рад са овако имплементираним скупом:

- [5 поена] Формирање празног скупа
- [10 поена] Уметање новог елемента у скуп
- [5 поена] Провера припадности елемента скупу
- [10 поена] Форматиран испис садржаја стабла који представља скуп
- [5 поена] Брисање скупа из меморије

Поред наведених операција, у зависности од редног броја проблема j који се решава, реализовати и следећу операцију за рад са два скупа представљена на овај начин [15 поена]:

0. Унију два скупа (елементи уније припадају барем једном од скупова)
1. Пресек два скупа (елементи пресека припадају оба скупа)
2. Разлику два скупа (елементи разлике скупа А и В припадају скупу А, али не и скупу В)
3. Симетричну разлику два скупа (елементи припадају тачно једном од скупова)

Стабло којим се представља скуп треба у сваком тренутку да буде балансирано према AVL критеријуму. Програм треба да чита скуп кључева који се умећу у стабло са стандардног улаза.

[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.

Напомена: Поени за интеракцију са корисником се **не добијају** уколико се само исписује мени, без одговарајућих позива функција у позадини. Код исписа стабла, испис неке комбинације обилазака стабла се не прихвата као адекватно решење, већ стабло треба адекватно форматирати.