

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2018-2019
- 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. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
3. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
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. **Примена рекурзије се неће признати као решење проблема које може освојити 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 низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
7. Одбрана првог домаћег задатка ће се обавити у **среду, 24.10.2018. и четвртак, 25.10.2018.** према распореду који ће накнадно бити објављен на сајту предмета.
8. Пре одбране, сви студенти раде тест знања за рачунаром коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да се пријаве на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико је студент регистрован на систем путем свог налога електронске поште на серверу mail.student.etf.bg.ac.rs.
9. Формуле за редни број **i** и **j** комбинација проблема коју треба користити приликом решавања првог и другог задатка, респективно, су следеће:
(**R** – редни број индекса, **G** – последње две цифре године уписа):
$$j = (R + G) \bmod 3$$
$$i = (R + G) \bmod 3$$
10. Имена датотека које се предају морају бити **dz1p1.c** и **dz1p2.c**
11. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

Задатак 1 – имплементација BST са понављањем кључева [50 поена]

Написати програм за рад са бинарним стаблом претраживања које дозвољава понављање кључева. У стабло се смештају целобројни кључеви. Потребно је подржати операције формирања стабла, претраге стабла, уметање целобројних кључева у стабло, испис садржаја стабла, брисање кључева и брисање стабла. У зависности од редног броја проблема који се решава, уметање поновљених кључева реализовати на следећи начин:

0. Уметањем поновљеног кључа као десно дете
1. Уметањем поновљеног кључа као лево дете
2. Коришћењем бројача који за сваки чвор памти број понављања одговарајућег кључа

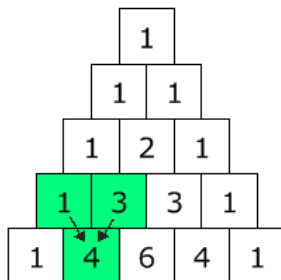
Поред наведених операција, у зависности од редног броја проблема који се решава, реализовати и следећу операцију:

0. Налажење другог најмањег кључа
1. Одређивање кључа са највећим бројем понављања
2. Испис крајњег десног кључа на задатом нивоу

Демонстрацију рада програма остварити кроз једноставан интерактивни мени који ће омогућити комуникацију са корисником.

Задатак 2 – претрага линеарних структура података [50 поена]

Написати програм за поређење перформанси различитих техника претраге задатог реда Паскаловог троугла (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.



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

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

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

У зависности од редног броја проблема који се решава, реализовати следећу комбинацију техника претраживања, као и њихову хибридную имплементацију:

0. Бинарно, интерполационо, хибридна имплементација
1. Тернарно, интерполационо, хибридна имплементација
2. Бинарно, тернарно, хибридна имплементација

Код хибридне имплементације, потребно је осмислити начин како најбоље комбиновати дате две основне технике или њихове варијанте, тако да се добију што боље перформансе претраге.

Демонстрацију рада програма остварити кроз једноставан интерактивни мени који ће омогућити комуникацију са корисником. Треба омогућити унос редног броја реда Паскаловог троугла за претраживање, унос кључа за претраживање, генерисање секвенце кључева за претраживање, покретање симулације и испис резултата. Програм написати тако да може да чита улазне податке (редни број реда Паскаловог троугла који треба претраживати и кључеве на које се врши претрага) са стандардног улаза или из датотека. Резултате исписати на главном излазу.