Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

ТЕМА ЗА ПРОЕКТ

към курс “Структури от данни и програмиране”
за специалност Компютърни науки,
зимен семестър 2019/2020 г.

Йерархия от контейнери

Да се реализира хетерогенен свързан списък от състоящ се от **n** контейнера. Числото **n** се въвежда от потребителя. Всеки контейнер може да е шаблон на двусвързан списък, стек или опашка и трябва притежава операция `bool member(T const& x)`, която проверява дали даден елемент **x** се среща в контейнера.

За хетерогенния списък да се реализира следната функционалност:

1. Нека е даден файл с **n** реда от цели числа. Да се реализира извличане на всеки ред от файла и въвеждане на числата в него в съответен контейнер. Числата предназначени за един контейнер се намират на един ред, а в началото на реда е отбелязан вида на съответния контейнер (`D`, `S` или `Q`).
2. За всеки от контейнерите да се поддържат подходящи функции за включване и изключване на елементи.
3. Да се реализира функция, която проверява дали даден елемент се среща в някой от контейнерите в хетерогенния списък.
4. Да се реализира функция за добавяне на елемент към хетерогенния списък реализираща равномерно разпределение на елементите (**load balancing)** — елементът да се добавя към контейнера с най-малка големина.
5. Да се напише функция, която проверява дали в контейнер се среща елемент, отговарящ на дадено условие, като условието е дефинирано като `using Condition = bool (*)(T const&);`
6. Да се напише функция за филтрация, която изтрива от всички контейнери в хетерогенния списък всички елементи отговарящи на дадено условие.
7. Да се напише функция, която сортира (поотделно) всички контейнери в хетерогенния списък. Могат да бъдат използвани помощни структури, ако е нужно.
8. Да се реализира функция визуализираща съдържанието на хетерогенния списък по подходящ начин.
9. Да се реализира подходящ итератор, който обхожда елементите на вече сортираните контейнери от хетерогенния списък във възходящ ред, като прескача между контейнери когато е нужно така, че при обхождане да се получи монотонно растяща редица от числа.
10. Да се реализира функция, която търси даден елемент **x** във вече сортираните контейнери в хетерогенния списък и връща итератор към първата позиция на

която се среща елементът или невалиден итератор, ако в списъка няма контейнер, в който да има такъв елемент.

11. Да се напише функция, реализираща обратното записване във файл на променения хетерогенен списък, в същия формат, който се използва за извличането му.

Бонуси:

1. Да се реализира подходящ итератор, който обхожда всички елементи на всички контейнери в хетерогенния списък, като поддържа два режима на обхождане - обхожда елементите контейнер по контейнер, като във всеки контейнер елементите се обхождат подред или обхожда първо всички елементи стоящи на първа позиция в контейнерите (като контейнерите се обхождат според реда им в хетерогенния списък), след това елементите на втора позиция и т.н. Ако в даден контейнер няма достатъчно елементи, той се прескача на съответната стъпка от обхождането.
2. Да се реализира възможност всеки от шаблонните контейнери да е инстанциран от различен базов тип данни (например **bool**, **char**, **short**, **int**, **long long**, **char*** и т.н.) и да се модифицират извличането и записването от/в файла по подходящ начин, така че описанието на всеки контейнер да включва и описание на типът данни който съдържа.
3. Да се модифицират функциите за търсене, сортиране и филтриране, така че да разглеждат само контейнерите със тип на елементите, съответстващ на типа на аргумента на функцията.
4. Към възможните контейнери да се добави двоично дърво за търсене със следните уточнения:
    - В случая на двоично дърво за търсене, функцията за сортиране (точка 7) да балансира дървото (евентуално с използване на помощна структура)
    - Итераторът, който трябва да обхожда елементите на контейнерите в хетерогенния списък (точки 9 и 10) в случая на дърво трябва да позволява местене “напред” в дървото следвайки схема на обхождане “ляво-корен-дясно”