

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА
САМОПОДГОТОВКА
ПО
Структури от данни и програмиране
Контейнери и итератори

email: kalin@fmi.uni-sofia.bg

17 октомври 2018 г.

Следните задачи да се решат като упражнение за директно боравене с възлите на линеен двусвързан списък. Функциите да се тестват с подходящи тестове.

1. Да се дефинира функция `int count(dllnode<T>* l, int x)`, която преброява колко пъти елементът `x` се среща в списъка с първи елемент `l`.
2. Функция `dllnode<int>* range (int x, int y)` която създава и връща първия елемент на списък с елементи $x, x + 1, \dots, y$, при положение, че $x \leq y$.
3. Да се дефинира функция `removeAll (dllnode<T>*& l, const T& x)`, която изтрива всички срещания на елемента `x` от списъка `l`.
4. Да се дефинира функция `void append(dllnode*<T>& l1, dllnode<T>* l2)`, която добавя към края на списъка l_1 всички елементи на списъка l_2 . Да се реализира съответен оператор `+=` в класа на списъка.
5. Да се дефинира функция `dllnode* concat(dllnode<T>* l1, dllnode<T>* l2)`, който съединява два списъка в нов, трети списък. Т.е. `concat(l1, l2)` създава и връща нов списък от елементите на l_1 , следвани от елементите на l_2 . Да се реализира съответен оператор `+` в класа на списъка.

6. Да се дефинира функция `reverse`, която обръща реда на елементите на списък. Например, списъкът с елементи 1, 2, 3 ще се преобразува до списък с елементи 3, 2, 1.
7. Да се напише функция `void removeduplicates (dllnode *&l)`, която изтрива всички дублиращи се елементи от списъка *l*.

Следните задачи да се решат като упражнение за работа с итератори. Функциите да се тестват с подходящи тестове върху двата вида контейнери. Има ли разлика в производителността за някои от тях?

8. Да се разшири итераторът на динамичния масив така, че да поддържа оператора за стъпка назад `--`.
9. Да се дефинира функция `map`, която прилага едноаргументна функция $f : int \rightarrow int$ към всеки от елементите на произволен контейнер. Да се дефинира и шаблон на функцията за списък с произволен тип на елементите.
10. Да се напише функция `bool duplicates (...)`, която проверява дали в контейнер има дублиращи се елементи.
11. Да се напише функция `bool issorted (...)`, която проверява дали елементите на даден контейнер са подредени в нарастващ или в намаляващ ред.
12. Да се напише функция `bool palindrom (...)`, която проверява дали редицата от елементите на даден контейнер образува палиндром (т.е. дали се чете еднакво както отляво надясно така и отдясно наляво).