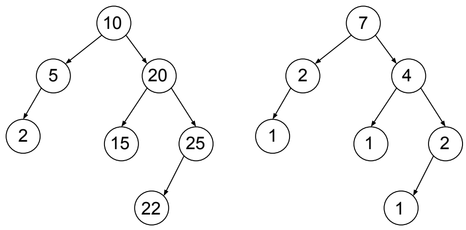
Задачи за задължителна самоподготовка
по
Структури от данни и програмиране
Двоични дървета 3

email: kalin@fmi.uni-sofia.bg

8 ноември 2017 г.

1. Дадено е двоично дърво. Да се напише функция, проверяваща дали има поне две различни нива от дървото, чиито множества от елементи съвпадат.

2. Да се реализира метод `bool BTree<T>::isBOT()`,който проверява дали двоичното дърво е наредено.

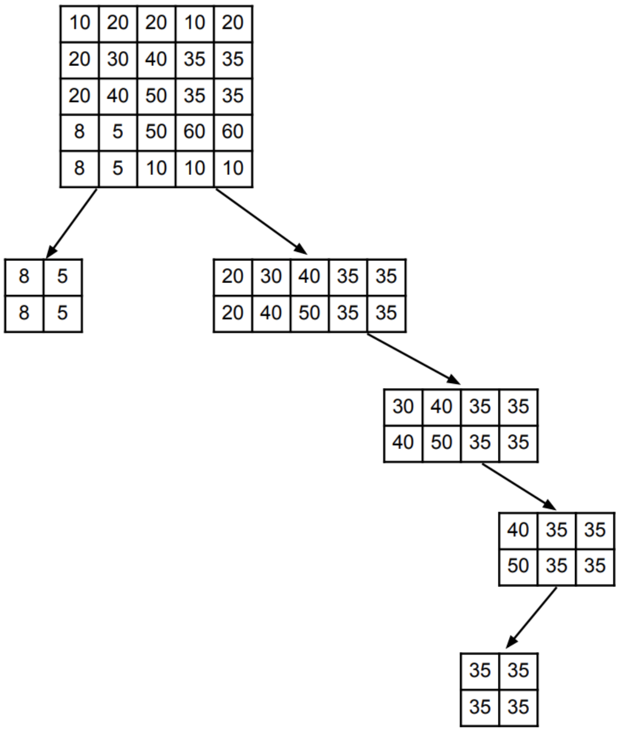


Фигура 1. Примерно дърво и същото дърво, стойностите на чиито възли са заместени с размера на съответното им поддърво.

3. Стойността на всеки възел `V` в дадено двоично дърво от числа да се замени с броя на всички елементи на поддървото, на което `V` е корен. Вж. фигура 1. При операцията всеки от възлите да бъде посетен най-много веднъж.

4. Нека е дадена матрица от цели числа $A_{M\times N}$ с елементи $(a_{i,j})$. "Лява" подматрица на A наричаме такава подматрица $A'_{M'\times N'}$ на A,

всеки елемент на която е по-малък от $a_{0,0}$. "Дясна" подматрица на A наричаме такава подматрица $A'_{M' \times N'}$ на A, всеки елемент на която е по-голям от $a_{0,0}$.



Фигура 2. Наредено дърво от матрици.

По матрицата A да се построи двоично дърво T със следните свойства:

- Коренът на T съдържа матрицата A.
- Нека v е произволен възел от дървото T, съдържащ матрица X. Ако X има поне една лява подматрица с размер поне 2×2, то левият наследник на X съдържа най-голямата (по брой елементи) лява подматрица на X. Ако има повече от една лявва подматрциа с максимален брой елементи, то левият наследник на v е произволна една от тях. Ако X няма лява подматрица с размер поне 2×2, то v няма ляв наследник.

- Аналлогичното свойство за десния наследник на v и най-голямата дясна подматрица (подматрици) на X.

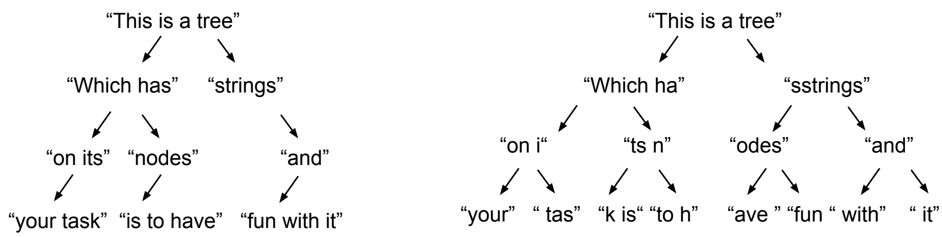
На фигура 2 е изобразено едно такова дърво.

(а) Да се избере подходящо представяне на матрици и на двоично дърво с матрици по върховете.

(б) Да се дефинира функция за построяване на дърво по горното правило по дадена матрица за корена му.

(в) Да се отпечата дървото чрез Graphviz. Повече информация за отпечатване на матрици като елемент на дървото може да се намери в документацията на Graphviz.

5. Дадено е дърво с низове по върховете. Дървото да се балансира по следния начин:



Фигура 3. Примерно дърво от низове преди и след балансирането.

(а) Резултатното дърво има същия брой нива като изходното.

(б) Всяко k-то ниво на резултатното дърво да съдържа точно 2^k елемента (считаме, че коренът е на ниво 0).

(в) Нека s_k е низът, получен при конкатенацията на всички низове на ниво k на изходното дърво, обхождани от ляво надясно. Нека дължината на низа s_k е n_k символа. i-тият пореден елемент на нивото k в резултатното дърво да съдържа i-тата поредна последноватeлност от $\lceil n_k/2^k \rceil$ на брой символи на s_k, освен най-десния, който съдържа последните “останали” символи от s_k. Т.е. s_k да се “раздели” поравно между елементите в резултатното дърво.

На фигура 3 са илюстрирани примерно изходно дърво и резултатът от балансирането му по горното правило. Всички елементи на ниво 1, освен последния, съдържат по $8 = \lceil 16/2 \rceil$ символа. Всички елементи на ниво 2, освен последния, съдържат по $4 = \lceil 14/4 \rceil$ символа и т.н.

Упътване: предварително намерете вектора $(s_0, s_1, ..., s_{h-1})$ *и го използвайте за балансирането.*