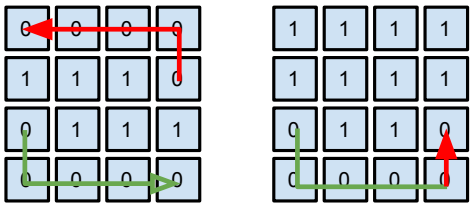
Задачи за задължителна самоподготовка #8

по

Структури от данни и програмиране

email: kalin@fmi.uni-sofia.bg

28 януари 2015 г.



Фигура 1а и 1б. Примерени лабиринти

1. Нека е дадена квадратна матрица от цели числа $N \times N$, представяща "лабиринт". Елементи на матрицата със стойност 0 смятаме за "проходими", а всички останали - за "непроходими". Път в лабиринта наричаме всяка последователност от проходими елементи на матрицата, които са съседни вертикално или хоризонтално.

   Да се дефинира функция `bool downstairs (int sx, int sy, int tx, int ty)`, която проверява дали съществува път от елемента (sx, sy) до елемента (tx, ty), такъв, че всеки следващ елемент от пътя е или вдясно, или под предишния. Такъв път да наричаме "низходящ".

   Пример: На фигура 1а такъв път съществува от елемента $(0, 2)$ до елемента $(3, 3)$, но не и от $(3, 1)$ до $(0, 0)$.

2. При условията на дефинициите от предишната задача, да се дефинира функция `bool connected()`, която проверява дали от всеки елемент на матрицата (sx, sy) до всеки елемент на матрицата (tx, ty), такива, че $sx \leq tx$ и $sy \leq ty$, съществува низходящ път.

   Пример: За лабиринта от фиг. 1а условието е изпълнено, но не и за лабиринта от фигура 1б.

3. Да се дефинира метод `int BTree::searchCount (bool (*pred)(const T&))` към клас `BTree`, който намира броя на елементите на дървото, които "удовлетворяват" предиката `pred`.

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

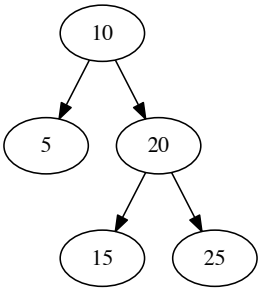
5. Да се дефинира метод `BTree::level (int k)`, който намира и връща вектор, съдържащ стойностите на всички елементи на дървото, които са на ниво k (т.е. има път от корена до тях с дължина k)

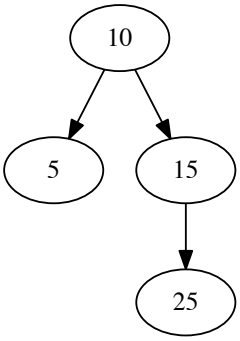
6. Да се дефинира метод `BTree BTree:without (int x)`, който построява копие на двоично наредената дървото, но без елемента x (т.е. "изтрива x").

   Пример: На фигура 2а е показано двоично наредено дърво, а на фигура 2б - негово копие без елемента 20.

7. Да се дефинира оператор `T BTree::operator[](int i)`, който намира i-тият пореден елемент на дървото при обхождане ляво-корен-дясно.

   Пример: За дървото от фигура 2а елементът с пореден номер 0 е 5, с номер 1 е 10, с номер 2 е 15 и т.н.





(а) Фигура 2. Двоично наредено дърво (б) Фигура 2. Копие без елемента 20