

Теоретический материал

Подробные материалы о всех видах сбалансированных деревьев и операциях с ними (добавление и удаление элементов) смотрите в слайдах лекции 2

АВЛ-деревья

АВЛ-дерево — сбалансированное по высоте двоичное дерево поиска: для каждой его вершины высота её двух поддеревьев различается не более чем на 1. Для АВЛ-деревьев **сбалансированность определяется разностью высот правого и левого поддеревьев любого узла**. Если эта разность по модулю не превышает 1, то дерево считается сбалансированным. Данное условие проверяется после каждого добавления или удаления узла, и определен минимальный набор операций перестройки дерева, который приводит к восстановлению свойства сбалансированности, если оно оказалось нарушено.

Для каждого узла дерева в структуру вводится переменная – показатель баланса (разность высот правого и левого поддеревьев)

Структура данных, описывающая узел АВЛ-дерева, имеет вид:

```
Struct Node{  
int key;  
Node * left;  
Node * right;  
Node * parent;  
int balans;  
};
```

Для упрощения работы алгоритмов в структуру допускается ввести указатель на родительский узел.

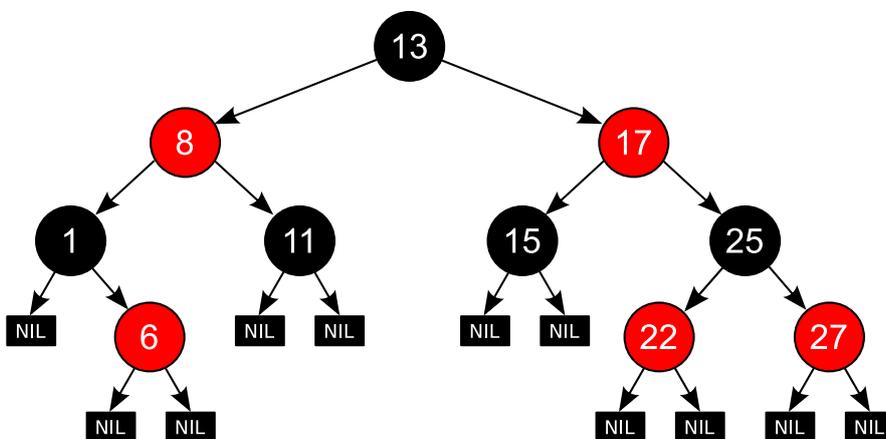
Красно-черные деревья

Красно-черное дерево (англ. red-black tree, RB tree) — один из видов самобалансирующихся двоичных деревьев поиска, гарантирующих логарифмический рост высоты дерева от числа узлов и позволяющее быстро выполнять основные операции дерева поиска: добавление, удаление и поиск узла. Сбалансированность достигается за счёт введения дополнительного атрибута узла

дерева — «цвета». Этот атрибут может принимать одно из двух возможных значений — «чёрный» или «красный».

Свойства КЧ-деревьев:

1. каждый узел либо красный, либо черный;
2. каждый лист (фиктивный) – черный;
3. если узел красный, то оба его сына – черные;
4. все пути, идущие от корня к любому фиктивному листу, содержат одинаковое количество черных узлов;
5. корень – черный.



Структура данных, описывающая узел красно-черного дерева, имеет вид:

```
Struct Node{
int key;
Node * left;
Node * right;
Node * parent;
char color;
};
```

Для упрощения работы алгоритмов в структуру допускается ввести указатель на родительский узел.

Самоперестраивающиеся деревья (splay деревья)

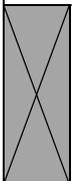
Самоперестраивающееся дерево является двоичным деревом поиска, в котором поддерживается свойство сбалансированности. Это дерево принадлежит классу «саморегулирующихся деревьев», которые поддерживают необходимый баланс

ветвления дерева, чтобы обеспечить выполнение операций поиска, добавления и удаления за логарифмическое время от числа хранимых элементов. Это реализуется без использования каких-либо дополнительных полей в узлах дерева (как, например, в Красно-чёрных деревьях или AVL-деревьях, где в вершинах хранится, соответственно, цвет вершины и глубина поддеревя). Вместо этого «расширяющие операции» (splay operation), частью которых являются вращения, выполняются при каждом обращении к дереву.

Операция $splay(T, k)$ выполняется следующим образом. Сначала производится поиск узла с ключом k в дереве обычным способом, спускаясь вниз, начиная с корня. При этом запоминается пройденный путь. В итоге, получаем указатель на узел дерева либо с ключом k , либо с его предшественником или последователем, на котором закончился поиск. Далее, происходит возвращение назад по запомненному пути, с перемещением этого узла к корню. Для того, чтобы при этом сохранялись свойства двоичного дерева поиска, необходимы повороты.

Задание 1

Задача:



Создать двоичное дерево поиска (в узлах хранятся целые положительные числа). Рассчитать для каждого узла построенного дерева показатель баланса

Решение:

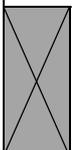


Ответ:



Задание 2

Задача:



Создать AVL-дерево (узлы добавляются как при стандартном алгоритме вставки узла, при нарушении баланса производится необходимая перестройка).

Решение:



Ответ:



Задание 3*

Задача:

Для AVL-дерева (построенного в задании 2) реализовать операции удаления узла дерева с учетом требования сохранения сбалансированности дерева после удаления узла.

Решение:



Ответ:



Задание 4

Задача:

Создать двоичное дерево поиска (в узлах хранятся целые положительные числа). Проверить для построенного дерева свойства красно-черного дерева.

Решение:



Ответ:



Задание 5*

Задача:

Создать красно-черное дерево (узлы добавляются как при стандартном алгоритме вставки узла, при нарушении свойств красно-черного дерева производится необходимая перестройка и перекрашивание). Реализовать операцию удаления узла из красно-черного дерева.

Решение:



Ответ:



Задание 6*

Задача:

Создать самоперестраивающееся дерево, для которого добавление нового узла и удаление существующего узла будет производиться на основе операции splay.

Решение:



Ответ:

