

Бинарные деревья поиска

Петрусович Денис Андреевич

Доцент кафедры Проблем управления, к.ф.-м.н.

e-mail: petrusevich@mirea.ru

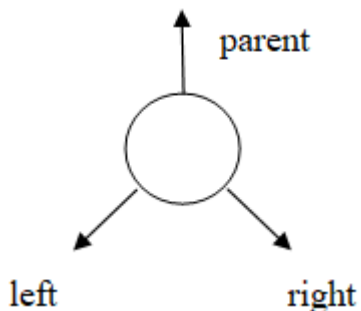
Основные поля элемента

Содержимое элемента: указатели на родительский элемент и два дочерних (левый и правый), информация в узле (data), высота узла в дереве (число узлов при спуске от корня до узла)

Класс дерева представляет из себя указатель на корень root

Можем перемещаться указателям left, right, parent

Признак листа (узел без сыновей): `left==NULL && right==NULL`

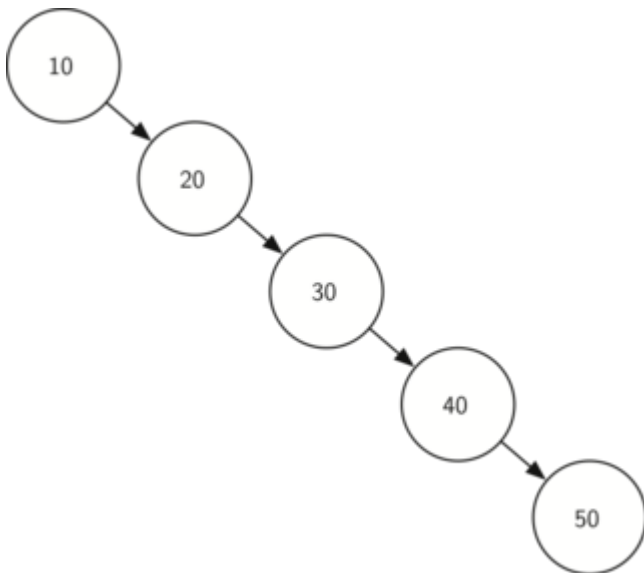


Сбалансированное дерево

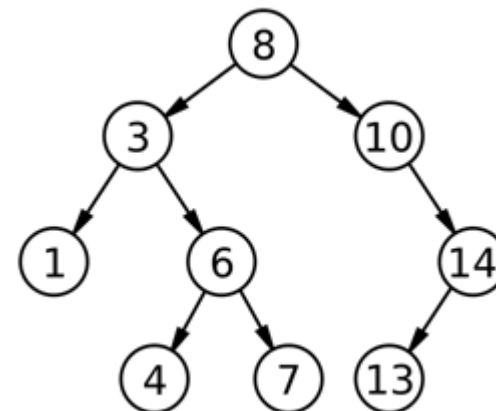
$|h(\text{left}) - h(\text{right})| < 2?$

Как оценить высоту сбалансированного дерева, зная число элементов n ?

Несбалансированное дерево
($h=5$)



Сбалансированное дерево
($h=4$)



Слайд 2

Дерево поиска

$\text{current} \rightarrow \text{left} \rightarrow \text{data} < \text{current} \rightarrow \text{data} < \text{current} \rightarrow \text{right} \rightarrow \text{data}$

Значение элемента слева меньше, чем в родителе

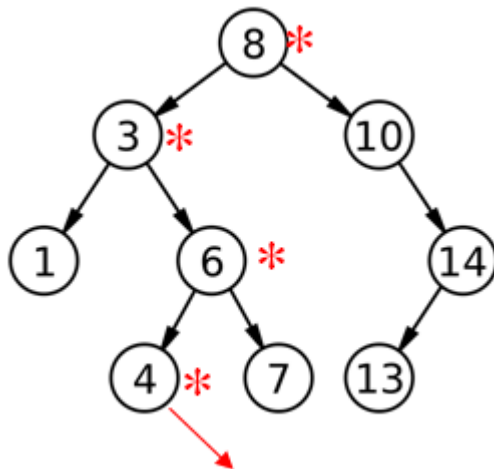
Значение элемента справа больше, чем в родителе

Поиск: спуск по дереву (каждый раз идем налево и направо, сравнивая $\text{current} \rightarrow \text{data}$ и искомое значение)

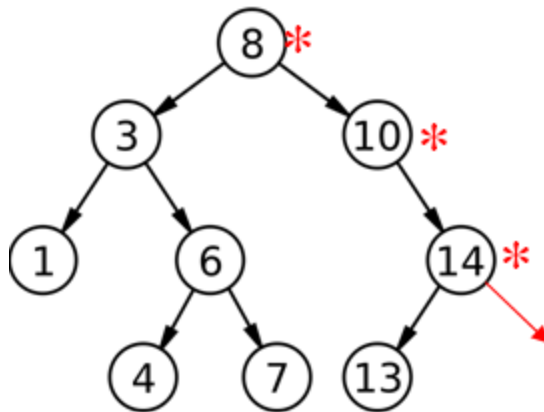
Добавление: поиск места для элемента (пока не найден элемент или пока не найдено свободное место)

Пример добавления элемента

Добавляем «5»

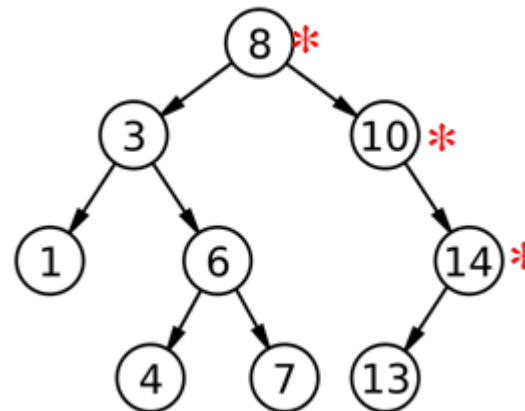


Добавляем «15»

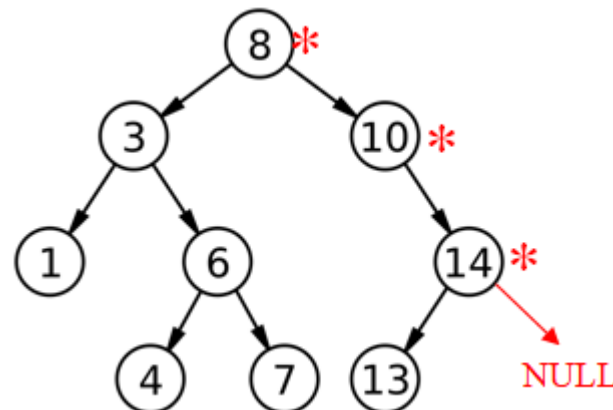


Пример поиска элемента

Ищем «14»



Ищем «15»



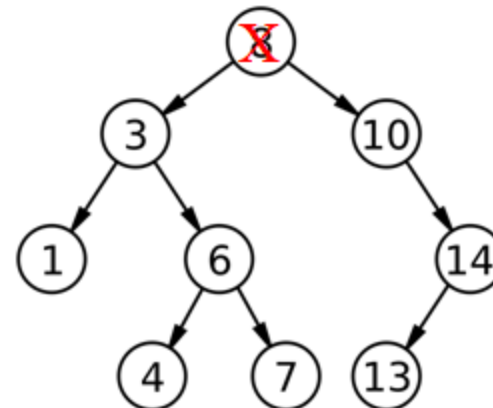
Где расположены min/max?

Как выполнить удаление?

Простые случаи: у удаляемого узла нет детей (отредактировать родителя), у удаляемого узла один ребенок (он занимает место удаляемого узла)

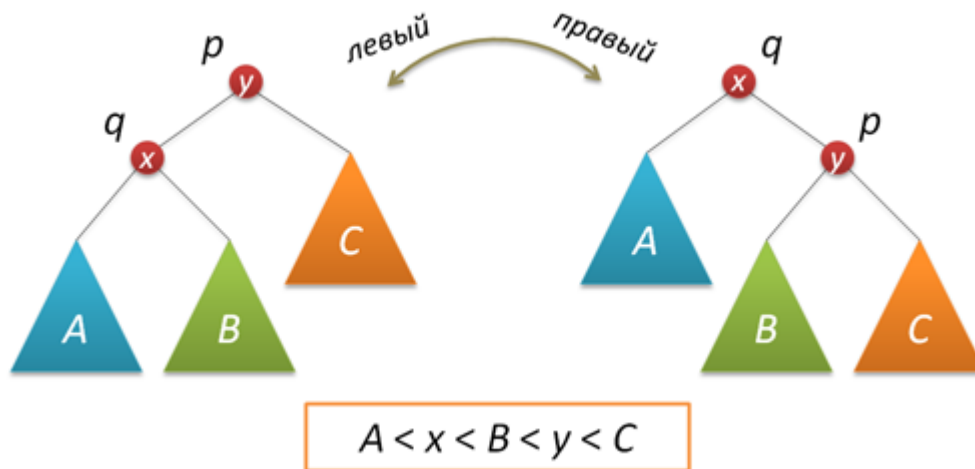
У удаляемого узла два ребенка. Какой узел займет его место?
Этот узел должен быть больше всех элементов слева, меньше всех элементов справа

Что это за элемент?



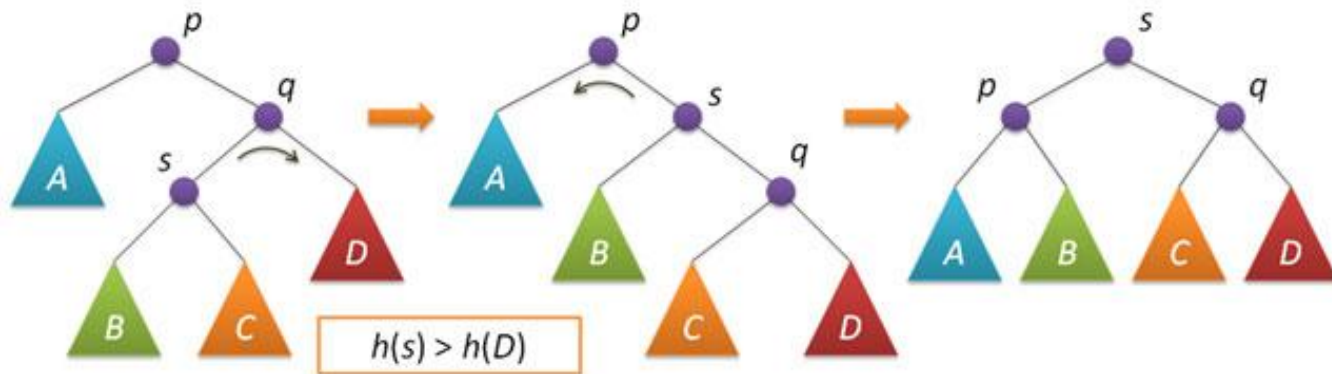
АВЛ-дерево (сбалансированное дерево поиска)

Повороты (свойства дерева поиска не нарушены, $p > q$):



АВЛ-дерево (сбалансированное дерево поиска)

Балансировка дерева – цепь поворотов (свойства дерева поиска не нарушены, $p < s < q$):

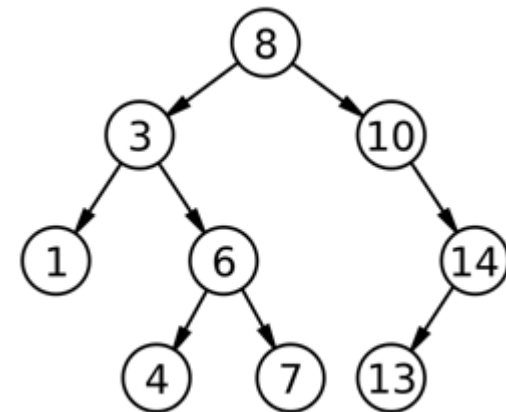


Задача LCA (наименьший общий предок, least common ancestor)

$$\text{LCA}(4, 7) = 6$$

$$\text{LCA}(4, 13) = 8$$

$$\text{LCA}(1, 4) = 3$$



Обходы дерева

Симметричный (InOrder)

Прямой (PreOrder)

Обратный (PostOrder)

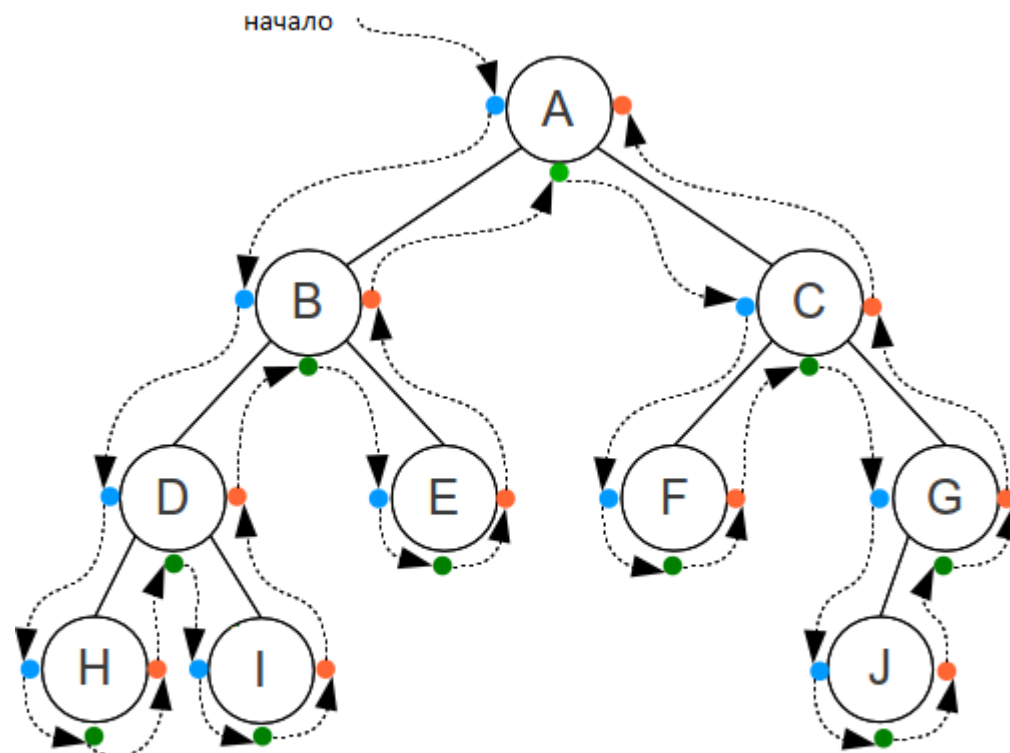
Для LCA (x, y):

1. Вывести элементы между

x, y в InOrder-обходе

2. Обратить внимание на

ИХ ВЫСОТЫ



прямой

ABDHECFGJ

симметричный

HDIBEAFCJG

обратный

HIDEBFJGCA

Слайд 10

Применение обходов: вычисление выражения с разными по приоритетам операциями

Построить дерево

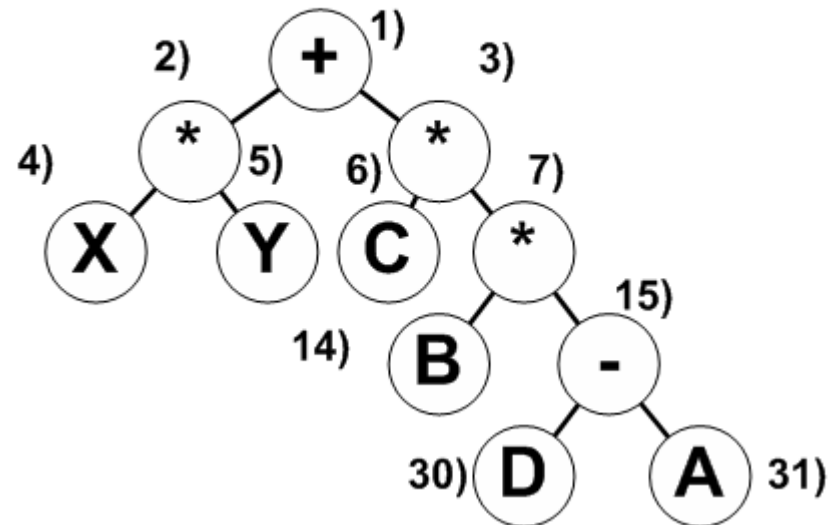
по выражению

Пройти дерево одним из

Обходов, вычисляя результат

Пример представления математического
выражения в виде дерева с помощью массива

$$Z = X * Y + C * B * (D - A)$$



Задачи

1. Учесть *parent во всех операциях
2. Удаление узла
3. LCA
4. $\text{Successor}(x)$ – элемент, который был бы в отсортированном массиве следующим за x
5. $\text{Predecessor}(x)$ – элемент, который был бы в отсортированном массиве предыдущим для x
6. Создать шаблон АВЛ-дерева

Список литературы

1. Кормен. Главы 12, 13
2. Вирт. Пункт 4.4
3. Топп, Форд. Главы 11, 13
4. Weiss. Пункты 4.3, 4.4
5. Das. Пункты 8.1 – 8.7, 8.13, 8.14
6. Лекции Data Structures. Unit 3; пункты 5.5.1, 5.5.2
7. Habr. AVL – деревья. <https://habr.com/ru/post/150732/>
8. Habr. Деревья поиска. <https://habr.com/ru/post/267855/>
9. ИТМО. Деревья поиска.
<https://neerc.ifmo.ru/wiki/index.php?title=%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D0%B0,%D0%BD%D0%B0%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F%D1%80%D0%B5%D0%B0%D0%B%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F>

Спасибо за внимание!