Самобалансирующиеся деревья поиска

Алексей Владыкин

СПбГУ ИТМО

11 марта 2011

- Сбалансированное дерево — дерево, в котором заполнены все уровни, возможно кроме последнего.
- Чем ближе дерево поиска к сбалансированному, тем эффективнее операции с ним: $O(\log_k n)$.
- Добавление или удаление элементов дерева поиска легко нарушает сбалансированность.
- Поддержание идеальной сбалансированности слишком дорого.
- На практике довольствуются «почти сбалансированными» деревьями.

- Рандомизация порядка вставки
- Декартово дерево со случайными приоритетами
- Вставка в корень с вероятностью

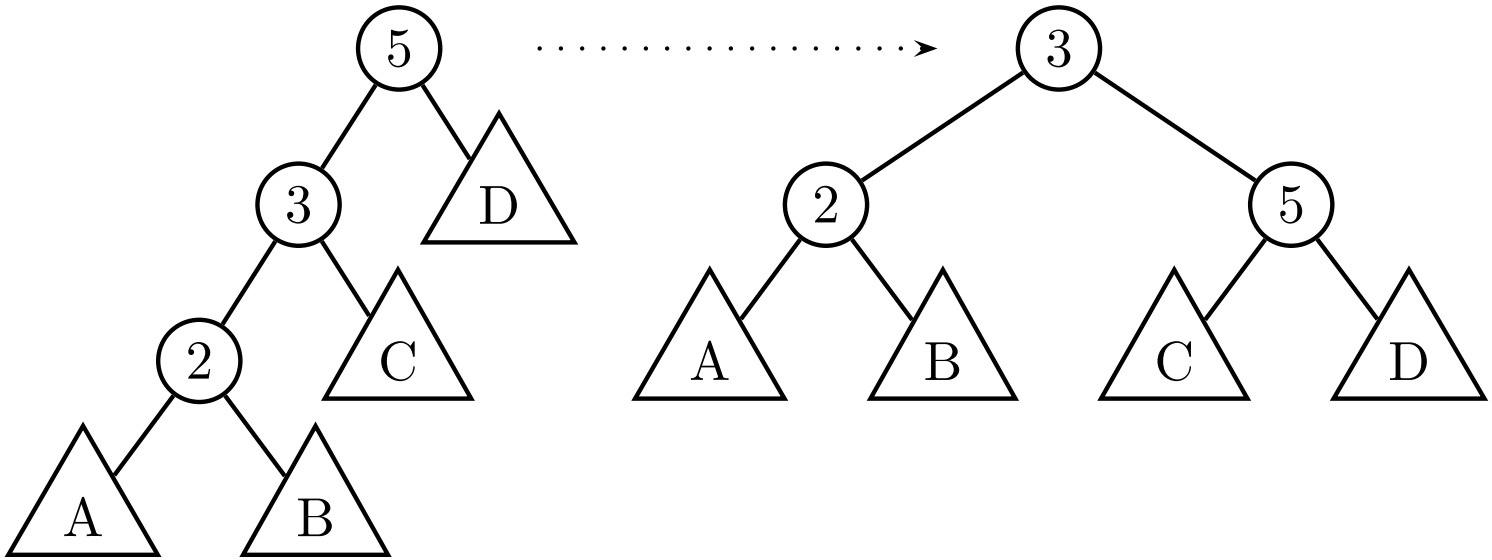
$$P = \frac{1}{N+1}, \quad \text{где } N \text{ — количество узлов в дереве}$$

- Адельсон-Вельский и Ландис, 1962 г.
- АВЛ-дерево — двоичное дерево поиска.
- Критерий сбалансированности: высоты левого и правого поддеревьев любого узла различаются не более чем на единицу.
- Показатель сбалансированности узла — высота правого поддерева минус высота левого поддерева.
- При любом числе узлов высота АВЛ-дерева превосходит высоту идеально сбалансированного дерева не более чем на 45%.
- Наихудший случай АВЛ-дерева — дерево Фибоначчи.
- Количество узлов в дереве Фибоначчи — числа Леонарда.

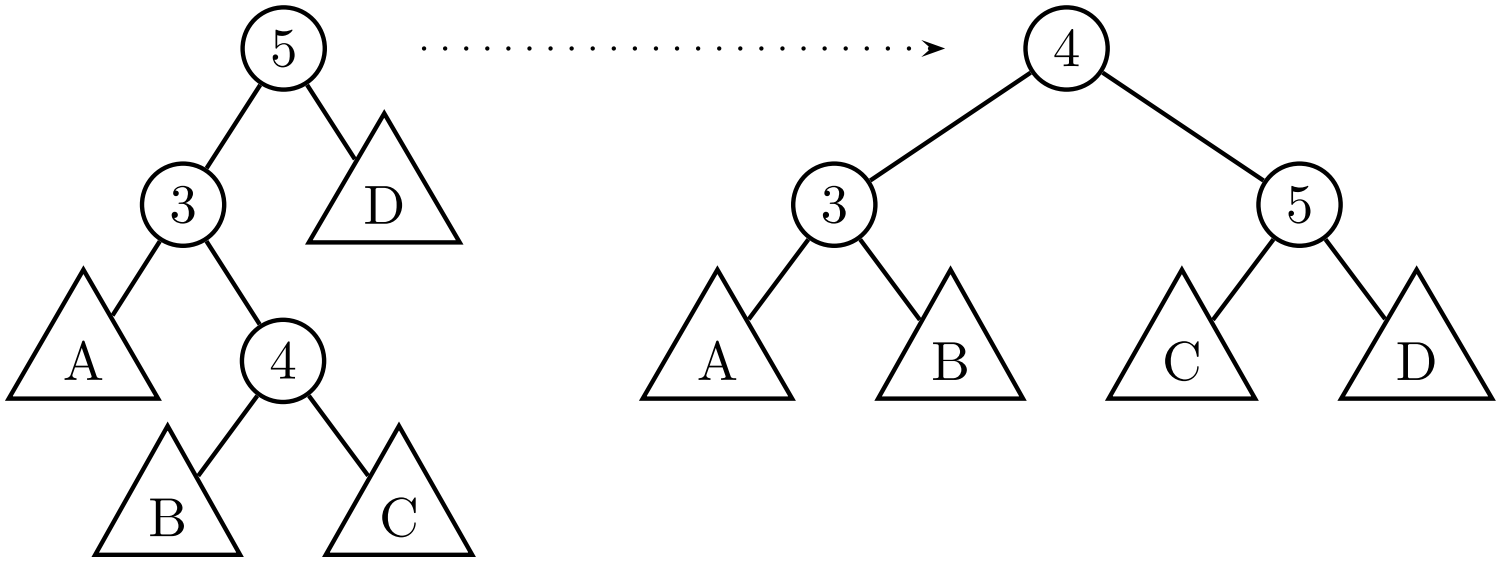
Балансировка

- После обычной вставки или удаления элемента проверяем все узлы на пути к корню на предмет нарушения балансировки (показатель сбалансированности ± 2).
- При нарушении сбалансированности применяем одно из четырех *вращений*: LL, LR, RR, RL.
- LL симметрично RR; LR симметрично RL.

LL-вращение



LR-вращение



- Rudolf Bayer, 1972 г. (symmetric binary B-tree)
- Leonidas J. Guibas и Robert Sedgewick, 1978 г.
- Красно-чёрное дерево — двоичное дерево поиска.
- Каждый узел помечен цветом: красным или чёрным.
- Определение:
  1. Все листья чёрные.
  2. Все потомки красных узлов чёрные.
  3. Чёрные высоты всех листьев равны.
- Следствие: высоты листьев красно-чёрного дерева отличаются не более чем в два раза.

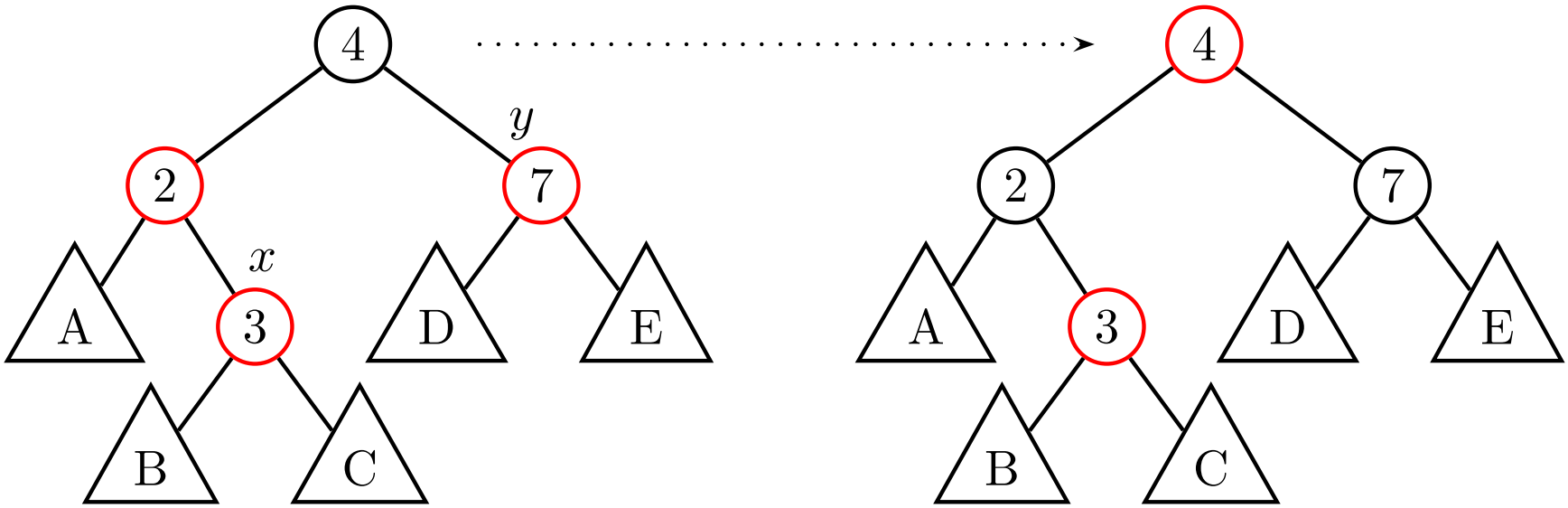
Добавление элемента

- Добавляем элемент как красный узел с двумя пустыми чёрными листьям.
- Могло нарушиться свойство (2), если родитель нового узла тоже красный.
- Возможны шесть случаев, три из них симметричны трём другим. Устраняем нарушение, двигаемся к корню, снова проверяем свойство (2).

- Удаление элемента — разобрать самостоятельно

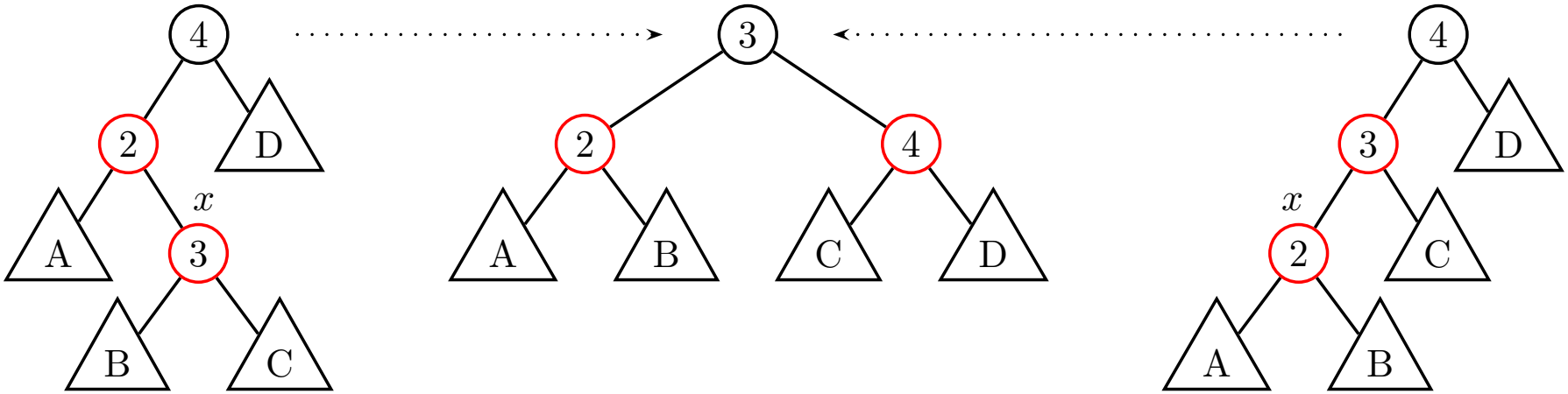
Добавление элемента

- Текущий красный узел x является ребёнком красного узла, его дядя y тоже красный.



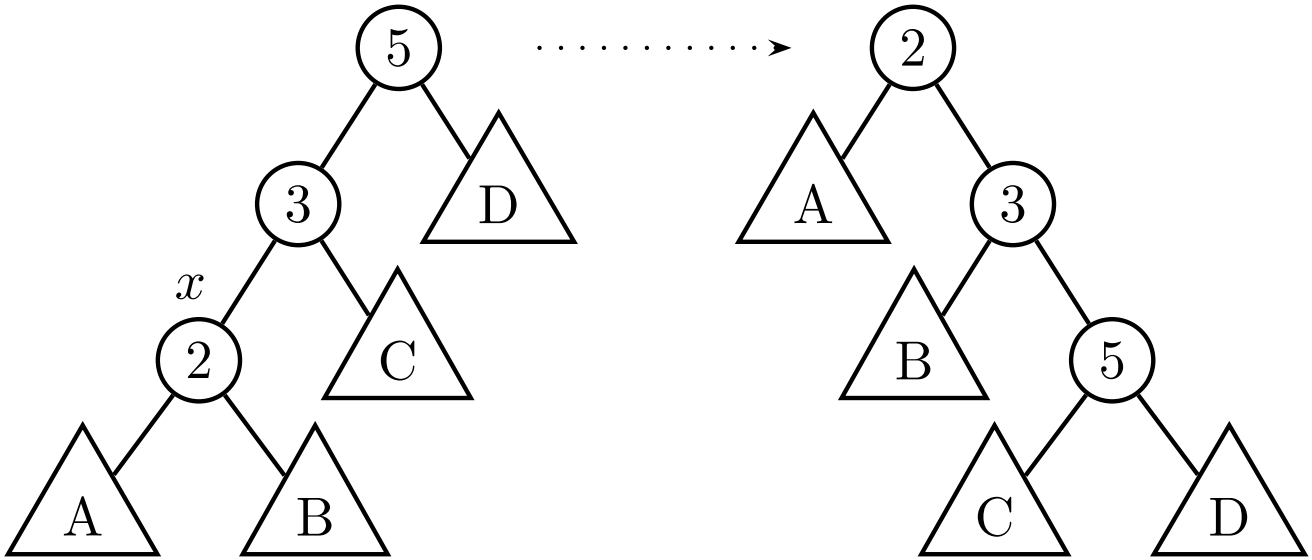
Добавление элемента

- Текущий красный узел x является правым или левым ребёнком красного узла, его дядя y чёрный.

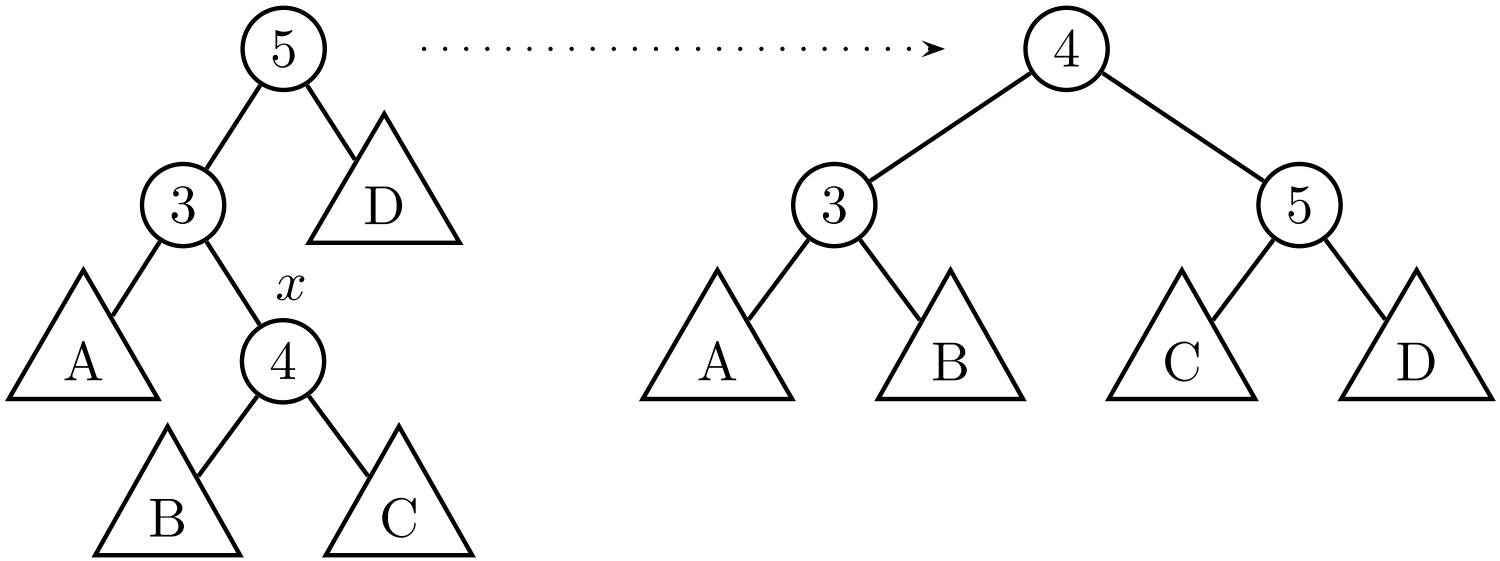


- Daniel Sleator и Robert Tarjan, 1985 г.
- После вставки *или поиска* элемент перемещается в корень путём серии вращений
- Трудоемкость M операций вставки или поиска в N-узловом дереве равна $O((M + N) \log(M + N))$.
- Это амортизированная оценка, т.е. трудоемкость отдельной операции в худшем случае может быть больше

- LL-вращение (RR — симметрично)



- LR-вращение (RL — симметрично)



- Rudolf Bayer и Edward McCreight, 1972 г.
- B-дерево — 2t-ичное дерево поиска ($t \approx 10^3$), предназначенное для хранения больших объемов данных во внешней памяти.
- Все листья находятся на одной и той же высоте.
- Число ключей в любой вершине, кроме корня, находится в диапазоне $t - 1 \leqslant k \leqslant 2t - 1$, где $t \geqslant 2$ — заданная минимальная степень B-дерева.
- В корне B-дерева может находиться от одного до $2t - 1$ ключей.

Добавление элемента

- Проход от корня к месту вставки
- Расщепление встретившихся по пути полных узлов
- Ключ-медиана расщепленного узла отправляется к родителю
- Рост возможен только за счет расщепления корня

- Удаление элемента — разобрать самостоятельно

Рекомендуемая литература

Вирт Н.

Алгоритмы и структуры данных: Пер. с англ. — 2-е изд., испр.
СПб.: Невский Диалект, 2001. — 352 с.: ил. // Глава 4

Кормен Т., Лейзерсон Ч., Ривест Р.

Алгоритмы: построение и анализ.
М.: МЦНМО, 1999. — 960 с., 263 ил. // Главы 14 и 19

Седжвик Р.

Фундаментальные алгоритмы на C.
К.: ДиаСофт ЮП, 2003. — 1136 с. // Главы 13 и 16