Тема 4. Быстрые деревья поиска

Лекция 10. AVL дерево

Обычное дерево поиска может вырождаться в список и соответственно трудоемкость работы может стать $O(N)$. Однако любой надор различных элементов можно организовать в дерево с логарифмической глубиной.

Проблема: можно ли организовать работу с деревом (поиск, добавление, удаление) с сохранением логарифмической сложности в любой момент времени?

Одно из решений — AVL дерево (Адельсон-Вельский и Ландис).

Идея — поддержка сбалансированности по глубине (длине) путем "поворотов" (иллюстрация).

Определение. Глубина (длина) дерева — это наибольшая длина его ветвей от корня.

Определение. Идеально сбалансированное дерево — это такое, в котором длины всех ветвей от корня отличаются не более чем на 1.

Очевидно, длина идеально сбалансированного дерева с N узлами есть $\sim \log_2 N$. Но поддерживать идеальную сбалансированность при добавлении и удалении не удается со сложностью $O(\log_2 N)$.

Определение. Сбалансированное AVL дерево — это такое, в котором в любой вершине длины левого и правого подеревьев отличаются не более чем на 1.

Пример. Идеально сбалансированное и AVL дерево — это разные вещи!

Определение. Баланс вершины есть разность длин правого и левого поддеревьев.

Пример. Дерево Фибоначчи — баланс всех внутренних вершин равен 1.

Утверждение. Длина дерева Фибоначчи с N вершинами есть $O(\log_2 N)$.

Доказательство. Пусть $N(k)$ есть количество вершин дерева Фибоначчи глубины k. Имеем рекуррентное соотношение

$N(k) = N(k-2) + N(k-1) + 1$

$N(0) = 0, \quad N(1) = 1$

Будем искать решением этого рекуррентного соотношения в виде

$N(k) = a^k - 1$

Действительно,

$$a^k - 1 = a^{k-2} - 1 + a^{k-1} - 1 + 1$$

или

$$a^2 - a - 1 = 0, \qquad a_1, a_2 = \frac{1 \pm \sqrt{5}}{2}$$

Умножение на константу не меняет решения, т.е. $C_1 a_1^k + C_2 a_2^k - 1$ — тоже решение Константы находятся из начальных условий:

$C_1 + C_2 - 1 = 0$

$C_1 a_1 + C_2 a_2 - 1 = 1$

Поскольку главным членом в элементах последовательности $N(k)$ является $C_1 a_1^k$, то при больших k можно считать, что $N(K) \sim C_1 a_1^k$, откуда получается, что

$$k \sim \log_2 N / \log_2 a_1 - \log_2 C_1 / \log_2 a_1 < \log_2 N / \log_2 a_1 < 1.5 \log_2 N$$

Утверждение. Произвольное AVL дерево длины k имеет не меньше узлов, чем дерево Фибоначчи длины k.

Доказательство легко получается по индукции.

Пусть утверждение верно для всех AVL деревьев длины от 0 до $k-1$. Рассмотрим корень произвольного AVL дерева длины k. Пусть это дерево имеет M узлов. Поддеревья этого корня имеют длины M_1 и M_2 не меньше $k-1$ и $k-2$. По предположению индукции имеем

$$M_1 \geq N(k-1), \quad M2 \geq N(k-2) \Longrightarrow M = M_1 + M_2 + 1 \geq N(k-1) + N(k-2) + 1 = N(k)$$

Утверждение. Длина произвольного AVL дерева с N узлами не превосходит $1.5 \log_2 N$.

Доказательство. Идея. Сопоставим дереву глубины k с N узлами точку на плоскости с координатами (k, N). По предыдущему утверждению все точки для AVL деревьев будут лежать не ниже кривой $N = N(k)$ (для дерева Фибоначчи). Кстати, эти точки будут лежать не выше кривой $N = 2^k - 1$. Таким образом, для данного N точки всех AVL деревьев будут образовывать горизонтальный отрезок между этими кривыми, т.е. максимальное k реализуется для дерева Фибоначчи.

Итак, AVL дерево имеет логарифмическую длину относительно числа узлов, и теперь надо обеспечить трудоемкость добавления и удаления с таким деревом в границах $O(\log N)$.