МФТИ, ФПМИ
Алгоритмы и структуры данных, осень 2021
Семинар №12. Деревья поиска, AVL-дерево

1. Дано подвешенное корневое бинарное дерево. Как за линейное время проверить, что оно является деревом поиска?
2. Как в дереве поиска искать максимальный и минимальный элементы? Как находить наименьший элемент, больший x, лежащий в дереве? Может понадобиться хранить дополнительное поле в каждой вершине.
3. Пусть $F_0 = 1, F_1 = 1$, а $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$. Пусть $\varphi = \frac{1+\sqrt{5}}{2}$. Докажите, что

$$F_{n-1} = \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}} \text{ при } n \geqslant 1.$$

4. Докажите, что при выполнении операции `insert` в AVL-дереве можно остановиться (не подниматься вверх), когда $\Delta(v) = 0$. Докажите, что если $\Delta(v)$ стало равным 1 или (-1), то это означает, что глубина поддерева с корнем в v (то есть $h(v)$) увеличилась. Докажите, что достаточно выполнить один поворот.
5. Докажите, что при выполнении операции `erase` в AVL-дереве можно остановиться (не подниматься вверх), когда $\Delta(v) = \pm 1$. Докажите, что если $\Delta(v)$ стало равным 0, то это означает, что глубина поддерева с корнем в v (то есть $h(v)$) уменьшилась. Всегда ли достаточно выполнить один поворот?
6. Пусть даны два AVL-дерева T_1 и T_2, причём все ключи T_1 из них строго меньше всех ключей T_2. Предложите алгоритм построения AVL-дерева, множество ключей которого совпадает с объединением множеств ключей T_1 и T_2, за время $O(\log(|T_1| + |T_2|))$.
7. В изначально пустое множество по одному добавляются или удаляются элементы. После выполнения каждого запроса сообщать медиану текущей версии множества. Асимптотика: $O(\log n)$ на запрос, где n — текущий размер множества.
8. Есть два множества отрезков на прямой, в них поступают запросы добавления. После каждого нужно сказать: сколько существует пар (отрезок из первого множества, отрезок из второго множества) таких, что они пересекаются? Асимптотика ответа на запрос: $O(\log n + \log m)$, где n, m — размеры множеств.
9. К изначально пустому множеству чисел S поступают запросы трёх типов: а) добавить x в S; б) удалить x из S; в) найти сумму элементов S, значения которых лежат в отрезке $[l, r]$; г) прибавить x ко всем элементам S. Отвечайте на каждый запрос за $O(\log q)$, а на запрос типа г) — за $O(1)$.

1. Достаточно научиться считать минимум и максимум в поддереве. Альтернативно, можно обойти дерева от корня, причём для каждой вершины выписать сначала её левое поддерево, потом её саму, а затем правое поддерево. Получившийся массив должен быть отсортирован.
2. Для поиска минимального элемента нужно спускаться влево, пока левый сын существует.
3. Воспользуйтесь индукцией по n.
4. Все утверждения лучше доказывать вместе: если алгоритм переходит из поддерева в наддерево, то это происходит только в случае, если в поддереве значение Δ было равно ± 1, а его глубина увеличилась (по сравнению с моментом времени до добавления элемента).
5. Решение аналогично решению задачи 3. Здесь, однако, не обойтись одним поворотом: их может быть много.
6. Пусть $h(T_1) \leqslant h(T_2)$. Удалите из T_1 максимальный элемент. Пройдите от корня T_2 вправо до той глубины, куда нужно подвесить T_1. Верните удалённый элемент, подвесьте к нему T_1 и необходимое поддерево T_2.
7. Храните два дерева поиска: список чисел, меньших медианы, и список чисел, больших медианы.
8. Проще считать количество пар непересекающихся отрезков.
9. В каждой вершине дерева храните сумму с поддерева. Также можно хранить отдельную константу, равную прибавляемому числу во всём дереве.