Домашняя работа №3

Шишацкий Михаил, 932

08.04.2020

Задача 6а. Множество поддерживающее операции, будет состоять из массива размером $log(N)$, в ячейках которого будут лежать декартовы деревья поиска по ключу - значение числа.

Запрос на *поиск* последователя будет выполняться следующим образом: Пройдём по массиву деревьев линейно и для каждого дерева найдем последователя элемента из запроса. По мере обхода будем сохранять значение минимального последователя. Асимптотика такой операции - $O(log^2(N))$ (т.к. в каждом дереве поиска мы умеем находить последователя за $O(log(N))$).

Запрос на *слияние* будем выполнять следующим образом: Если суммарное количество деревьев поиска в сливаемых множествах меньше $log(N)$, деревья из одного множества просто переносятся в свободные ячейки второго множества за $O(log(N))$. Если же деревьев больше или равно $log(N)$, сольём все деревья в одно дерево поиска за $O(log^3(N))$, так что амортизированная сложность выполнения всех запросов будет $O(log^2(N))$ (тяжёлое слияние выполняется в среднем через каждые $log(N)$ запросов слияния). Опишем алгоритм тяжёлого слияния:

По порядку будем сливать очередное дерево с уже слитыми: Посмотрим на корни двух деревьев k_1 и k_2. Без ограничения общности $k_1 < k_2$. Назовём левое поддерево меньшего корня T_1, правое поддерево большего корня - T_2. Справедливо неравенство $\text{AnyKey}(T_1) < k_1 < k_2 < \text{AnyKey}(T_2)$. Вызовем процедуру слияния рекурсивно для двух оставшихся поддеревьев, обозначим результат этой операции за S. Справедливо неравенство: $k_1 < \text{AnyKey}(S) < k_2$. Теперь ключи полученных деревьев упорядочены, и не пересекаются, их можно слить операцией *merge* за $O(log(N))$.

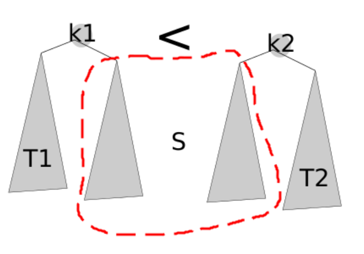

Рис. 1: слияние деревьев

Итого, асимптотика слияния - $O(log^3(N))$ (линейный проход по массиву деревьев $\times$ рекурсивное слияние $\times$ вызов *merge* на каждом уровне слияния).

Таким образом, мы научились обрабатывать любой запрос за $O(log^2(N))$ в среднем, поэтому q запросов обработаем за $O(q \cdot log^2(N))$ в среднем.

Некоторые тонкости: Для удобства, в вершинах деревьев будут лежать не сами числа, а указатели на вектор, где эти числа лежат. При этом все деревья будут двухсвязны, т.е. сыновья будут содержать указатель на родителя. Это нужно для того, чтобы во время вызова операции по элементу за $O(log(N))$ подняться от элемента к корню дерева, в котором он лежит, а оттуда восстановить текущее множество, в котором находится элемент.