

Деревья поиска. Задачи

Булгаков Илья, Гусев Илья

Московский физико-технический институт

Москва, 2023

Задача 1

Пусть в дополнение к стандартным запросам к дереву поиска добавляется ещё один: равновероятно выбрать один из ключей в дереве. Покажите, как реализовать такой запрос.

Задача 1 (Решение)

Пусть в дополнение к стандартным запросам к дереву поиска добавляется ещё один: равновероятно выбрать один из ключей в дереве. Покажите, как реализовать такой запрос.

Решение:

- Вместе с деревом поиска храним вектор всех ключей
- Элементы вектора и вершины дерева должны ссылаться друг на друга
- Когда удаляется один из ключей, можно поменять местами элемент вектора с последним и правильно пересчитать указатели

Задача 2

Поймите, как обрабатывается последовательность запросов `insert 1`, `insert 2`, \dots , `insert n` в splay-дереве.

Задача 2 (Решение)

Поймите, как обрабатывается последовательность запросов `insert 1`, `insert 2`, ..., `insert n` в `splay`-дереве.

Решение:

- После добавления числа i будет вызвана операция `splay`, сводящаяся к одному вызову `zig`
- В итоге получится бамбук влево: в корне находится i , в его левом сыне - $(i - 1)$, в его левом сыне - $(i - 2)$, и так далее.
- Каждая операция при этом выполняется за $O(1)$.

Задача 3

Покажите, что splay-дерево эффективно (линейно) работает, если использовать его как стек, то есть в качестве ключа при добавлении использовать только текущий размер дерева n , а удалять только ключ $n - 1$.

Задача 3 (Решение)

Покажите, что `splay`-дерево эффективно (линейно) работает, если использовать его как стек, то есть в качестве ключа при добавлении использовать только текущий размер дерева n , а удалять только ключ $n - 1$.

Решение:

- Удаляться в таком случае будет только корень, на его место подвешивается левый сын

Задача 4

Как сделать `merge` и `split` в `splay`-дереве? Обратите внимание, что времени обработки таких операций не должно превосходить время работы соответствующих операций `splay`.

Задача 4 (Решение)

Как сделать `merge` и `split` в `splay`-дереве? Обратите внимание, что времени обработки таких операций не должно превосходить время работы соответствующих операций `splay`.

Решение:

- Для реализации `merge` достаточно поднять в корень максимальный элемент левого дерева
- Для реализации `split` поднимем в корень максимальное число, не превосходящее ключа-разделителя x , тогда в качестве ответа нужно вернуть корень с левым поддеревом и правое поддерево.

Задача 5 (Неявное дерево поиска)

Задан массив a_1, \dots, a_m . К нему поступают запросы вида:

- а) Вставить x в позицию pos (то есть между двумя уже существующими элементами, нумерация смещается)
- б) Удалить число на позиции pos (нумерация смещается)
- в) Сообщить сумму на подотрезке массива

Обработайте каждый запрос за $O(\log n)$, где n — текущий размер массива.