Алгоритмы и структуры данных. Домашняя работа. Неделя 3

Автор: Петровский Александр M3139

Задание 1

Декартово дерево является бинарным деревом поиска по ключу, значит если в нем n вершин, то он может иметь различных (неизоморфных) корневых бинарных деревьев. . Различных перестановок приоритетов . Рассмотрим отношение . Для того, чтобы существовала пара перестановок, которые порождают одинаковые по форме деревья, по принципу Дирихле нужно, чтобы Q(n) > 1.

Так как при n > 4 Q(n) строго монотонна, то Q(n) всегда будут больше 1, а значит при n >= 3 найдется такая форма дерева, которой соответствует хотя бы две перестановки.

Задание 2

Рассмотрим числа, которые соответствуют символам строки, они имеют какие-то коды, пусть без потери общности это будет английский алфавит из маленьких букв, тогда их коды будут от 65 до 90. Рассмотрим строчку длины n. Минимальный хэш, который можно получить из такой строки равен 65 \* n, а максимальный – 90 \* n. Всего же строк длины n можно получить (90 – 65 + 1) ^ n = 26 ^ n. Понятно, что коллизий будет очень много. Также распределение будет неравномерное, так как хэш 65 \* n можно получить одним способом из строки длины n, 65 \* (n + 1) – n способами и т.д.

Итого: хэш действительно не очень

Задание 3

Воспользуемся принципом Дирихле. По условию дано, что ключей больше, чем n \* m, где m – число ячеек в хэш-таблице. По принципу Дирихле найдется n ключей таких, что их хэши совпадают, а значит именно эти ключи мы и выберем в качестве результатирующего набора, так как поиск мы осуществим за как максимум его размер.

Задание 4

Для листовых вершин утверждение тривиально. Рассмотрим вершину v и ее поддерево. Чтобы ключи дерева не образовывали непрерывного отрезка необходимо, чтобы в этом дереве отсутствовала некоторая не крайняя вершина. В случае поддерева v: пусть А – вершина поддерева v с минимальным ключом на нем, B – вершина поддерева v с максимальным ключом, K – какая-то не крайняя вершина поддерева v; т.е. выполняется условие, что A.key < K.key < B.key. Для того, чтобы отрезок был несвязный, вершина K должна отсутствовать в поддереве v, но значит она находится вне его, а это значит, что вершины v и K находятся в разных деревьях, а это в свою очередь значить, что A.key < B.Key < K.key или K.key < A.key < B.key в зависимости от расположения поддеревьев в дереве.