Алгоритмы и структуры данных. Домашняя работа. Неделя 1

Автор: Петровский Александр M3139

Задание 1

Обозначим последовательность из n данных различных целых чисел как . Отсортируем эту последовательность по возрастанию. Рассмотрим данное подвешенное двоичное дерево:

Заметим, что sz(T1) + sz(T2) + 1 = n. Рассмотрим корень дерева, обозначим его как . По свойству бинарного дерева поиска, все ключи в вершинах, которые находятся в левом поддереве (T1) вершины , меньше ключа в самой этой вершины, а все ключи в вершинах правого поддерева (T2) больше ключа в . Ключом вершины может оказаться только элемент , так как он больше sz(T1) элементов и меньше sz(T2) элементов последовательности .. Далее задача разбивается на одну или две подзадачи где работа ведется с двумя (одним) подмассивами и для левого и правого поддерева соответственно. В них аналогично выбирается элемент из последовательности в качестве ключа нового корня, поскольку ни один элемент из левого подмассива не может оказаться в правом поддереве, так как это нарушить свойство бинарного дерева поиска (аналогично для левого подмассива).

T1

T2

Задание 2

Очевидно, что минимальное количество вершин, которые могут добавиться к количеству листьев – 1. Действительно, возьмем просто одну вершину в качестве корня и проведем ребра ко всем m листьям. Докажем, что если дерево бинарное, то для заданного количества листьев при заданном условии структуры дерева, общее количество вершин в дереве будет максимальным. Рассмотрим . У такого дерева будет вершин. Заметим, что если мы сделаем какую-нибудь вершину n-детной (n > 2), то она заберет себе детей другой вершины, в итоге оставив ее без детей.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |

Бинарное дерево

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |

Измененное бинарное дерево (корень забрал детей красной вершины)

Пусть m не является степенью двойки. Тогда будем строить жадно: берем по две вершины на текущем уровне и создаем им одного родителя, пока не останется три вершины; если все вершины получили родителей, то переходим на новый слой, иначе осталось три вершины (или одна, но она и будет корнем, поэтому завершаем), которым даем одного родителя и переходим также новый слой.

Получаем:

**P.S.** Можно смотреть на задачу и иначе: Пусть у нас есть m листьев, тогда объединяя от 2 до m листьев мы получим от до 1 новых «листьев» соответственно. Исходя из того, что максимум мы можем добавить новых вершин, когда объединяем двух с одним родителем, получаем бинарное дерево, высота которого , а значит и количество вершин в этом дерево оценивается как .

Задание 3

Сперва заметим, что достигает равенства, когда это полное двоичное дерево. Действительно, пусть высота дерева h, листьев у такого дерева , тогда .

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Также в каждом бинарном дереве найдется уровень, который целиком заполнен. Поэтому в первоначальном дереве найдем такой уровень и разобьем дерево на поддеревья таким образом, чтобы вершины найденного уровня теперь являлись корнями своих поддеревьев. И так далее по этому принципу.

Пусть мы находимся на высоте h и попали в одну из следующих ситуаций:

1. Дерево-вершина (пень). Данный случай добавит в общую сумму .
2. Дерево с одним сыном. Данный случай добавит в общую сумму .
3. Дерево с двумя сыновьями. Данный случай добавит в общую сумму .

То есть в таких граничных случаях мы не можем получить . Понятно, что и на любой другой высоте k мы не сможем получить значение больше . Можно показать это при помощи индукции. База рассмотрена выше. Шаг индукции:

.

T1

T2

Задание 4

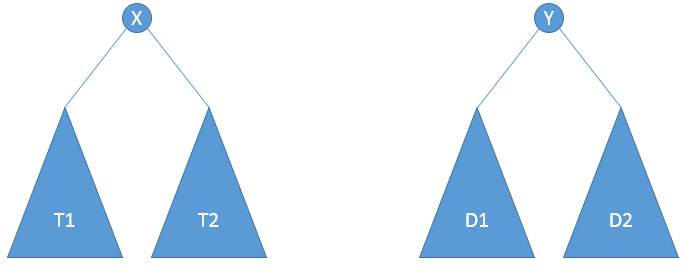
Одно из свойств красно-черного дерева – все пути, идущие от корня к любому листу, содержат одинаковое количество черных вершин. Значит, если высота дерева по черным вершинам от корня k, то максимальная высота может быть 2k, а минимальная – k. Также очевидно, что в корне может стоять элемент медианный элемент m = n / 2. Логично, что ключи, которые могут находится в корне находятся в каком-то промежутке [m – a; m + a], так как дерево мы можем отражать. Рассмотрим крайние случаи:

1. Длина пути в одном из поддеревьев k – 1 (корень черный), это значит, что все вершины в этом поддереве черные, а по свойству красного-черного дерева черные пути от любой вершины до любого листа поддерева этой вершины равны, а значит, что это будет полное бинарное дерево, то есть в нем вершина.
2. Длина пути в одном из поддеревьев 2\*k – 1 (корень черный), соответственно можно оценить максимальное количество вершин в нем: .

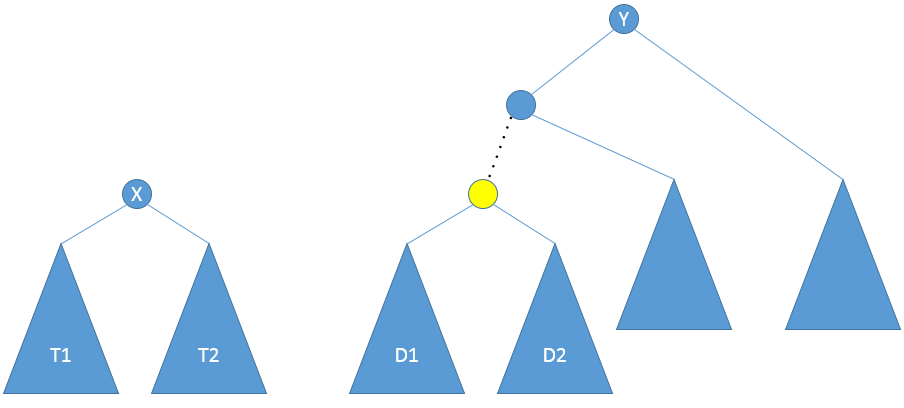
Получаем, что , т.е. .

Задание 5

Пусть у нас есть два AVL дерева T и D, их корни X и Y соответственно, пусть сперва их высоты равны , а .

**

За мы можем извлечь из дерева T вершину с максимальным ключом , и сейчас верно, что , а поскольку после удаления высота дерева T могла измениться максимум на единицу, то по условию для AVL дерева , то мы можем к слева подвесить T, а справа D.

Пусть у нас снова есть два AVL дерева T и D, их корни X и Y соответственно, а высоты не равны. Без потери общности пусть . **

Будем спускаться в дереве D по левым сыновьям, пока не окажется, что высота поддерева текущей вершины равна высоте дерева T, а такое будет, так как для каждой вершины выполняется свойство AVL-дерева. Обозначим эту вершину желтым цветом. Дальше, мы можем ее поддерево объединить с T, как мы делали выше. Высота получившегося дерева не изменится, поэтому мы можем подвесить получившееся дерево вместо желтой вершины, при этом все свойства AVL дерева D сохранятся.