

Анализ количества поворотов при добавлении перестановки из N элементов в АВЛ дерево

Дан массив, содержащий перестановку чисел от 0 до N . Найти среднее кол-во поворотов каждого типа. Провести численный эксперимент.

0.1. Введение в теорию

Двоичное дерево поиска – это двоичное дерево, для которого выполняются следующие дополнительные условия: У всех узлов левого поддерева произвольного узла значения ключей данных меньше, нежели значение ключа данных самого этого узла; У всех узлов правого поддерева произвольного узла значения ключей данных больше либо равны, нежели значение ключа данных самого этого узла.

Сбалансированное дерево – дерево, высота которого логарифмична кол-ву элементов.

АВЛ дерево – это бинарное дерево поиска, в котором для каждого элемента выполнено свойство: разница высот для левого и правого поддерева по модулю не превосходит 1. Данное условие является достаточным для сбалансированности дерева.

При стандартной вставке элементов в дерево поиска оно перестает быть сбалансированным. Для предотвращения этого используются следующие методы:

- 1) **Малый левый поворот**
- 2) **Большой левый поворот**
- 3) **Малый правый поворот**
- 4) **Большой правый поворот**

Повороты дерева - это операции, которые позволяют изменить структуру дерева не меняя порядка элементов. Используя их для соответствующих узлов, можно минимизировать высоту дерева, то есть сделать его сбалансированным.

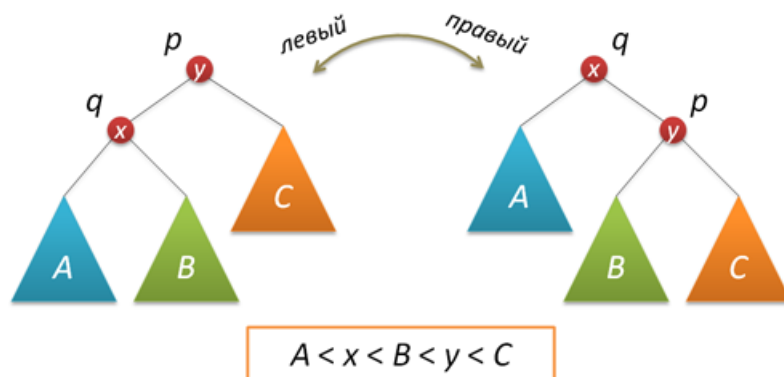


Рис. 1. Малые повороты. Разность высот поддеревьев по модулю равна 2, производим следующую трансформацию дерева

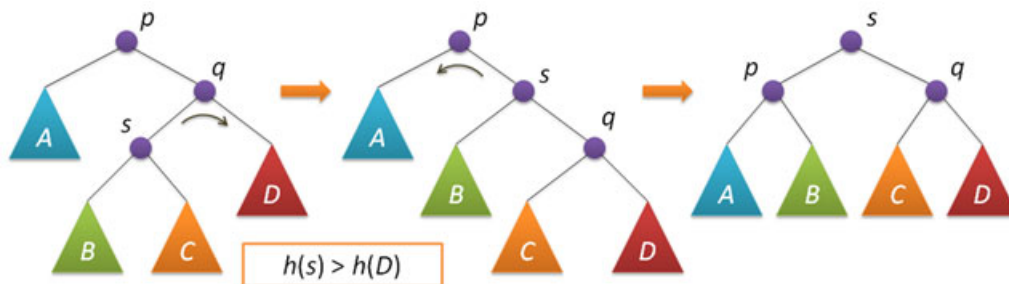


Рис. 2. Большой поворот применяется при условии $h(s) > h(D)$ и сводится в данном случае к двум простым – сначала правый поворот вокруг q и затем левый вокруг p

0.2. Эксперимент

Для проведения эксперимента была написана программа на языке C++ (файлы программы расположены в папке **programm_14** там же расположен тестовый файл **stdout.txt**, в котором расположен стандартный вывод программы). Она генерирует случайную перестановку из N элементов, записывает ее в АВЛ дерево поэлементно, при необходимости применяя повороты. При этом происходит подсчет их количества. Эта процедура повторяется несколько раз, после чего все значения усредняются.

Чтобы выявить зависимости эксперимент проводился для различных значений N , а именно для $N \in 500, 600, 700, \dots, 2000$. Главным образом это все было предназначено для анализа пропорциональности роста кол-

ва поворотов различного типа в зависимости от числа N .

0.3. Результаты и выводы

Вывод программы перенаправляется в файл **output.txt** в папке **documents**. По полученным результатам был построен график зависимости кол-ва различных поворотов от числа N . Эксперимент показал, что зависимость кол-ва поворотов каждого типа, от количества элементов перестановки с некоторой погрешностью линейна.

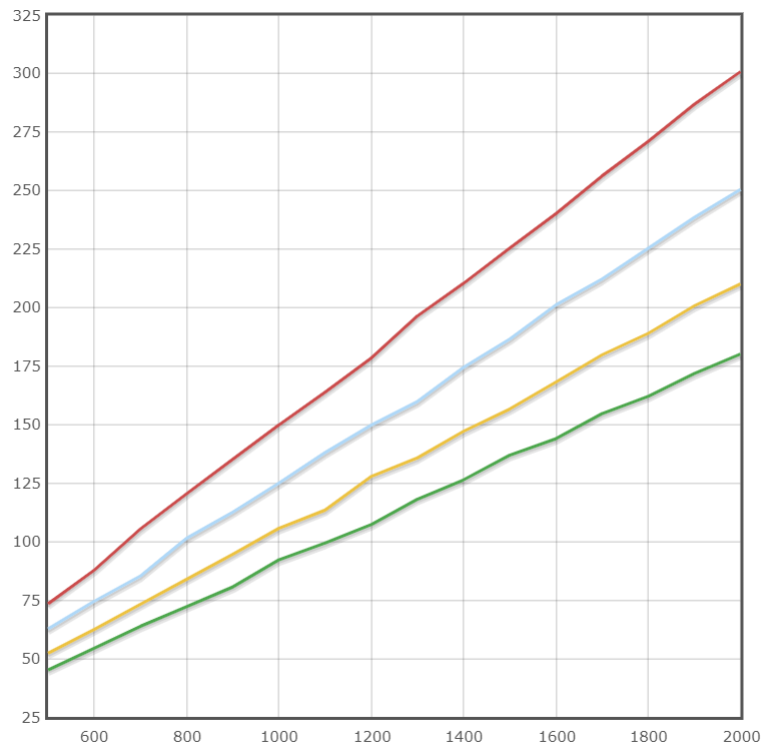


Рис. 3. Малый левый поворот – красная линия; Большой правый поворот – голубая линия; Большой левый поворот – желтая линия; Малый правый поворот – зеленая линия.

Несложно рассчитать коэффициенты пропорциональности:

- 1) Малый левый поворот : $k \approx 0,150$
- 2) Большой левый поворот : $k \approx 0,105$

3) Малый правый поворот : $k \approx 0,090$

4) Большой правый поворот : $k \approx 0,125$