**Домашнее задание 2. Федер Евгений, М3139**

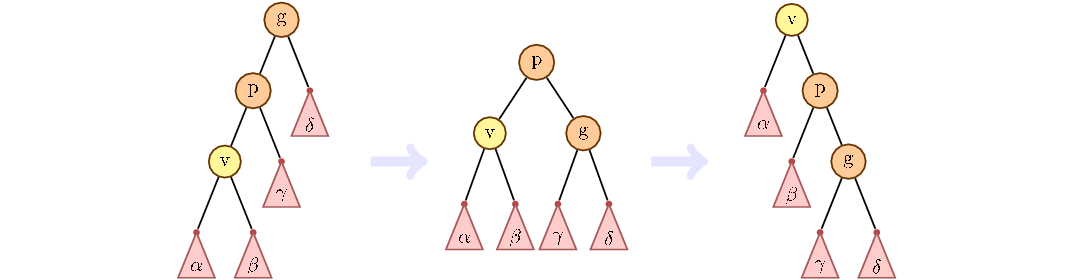
***Задание 1***

На лекции было доказано, что при операции zig амортизационная стоимость операции <= 3(rank(v`)-rank(v)) + 1. Еще мы знаем, что ранг любой вершины ограничен логорифмом. Так как у нас высота дерево может быть O(n), то получаем O(log(n)) + O(n) = O(n).

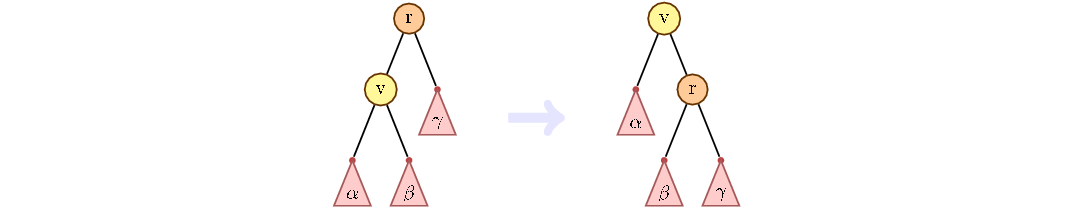
***Задание 2***

Такого дерева с более 7 вершинами не существует. Так как на каждой операции zig-zig, zig-zag и zig должен сохраняться инвариант, что результирующее поддерево x(вершины, который мы поднимаем) должно быть бамбуком. Очевидно, что этот инвариант не выполняется при zig-zag – там становится два поддерева из верщин которых мы поворачиваем.

При zig-zig если у нас есть поддеревья альфа и бета(а в общем случае они есть, кроме начала), то получится у v или p по два ребенка - плохо



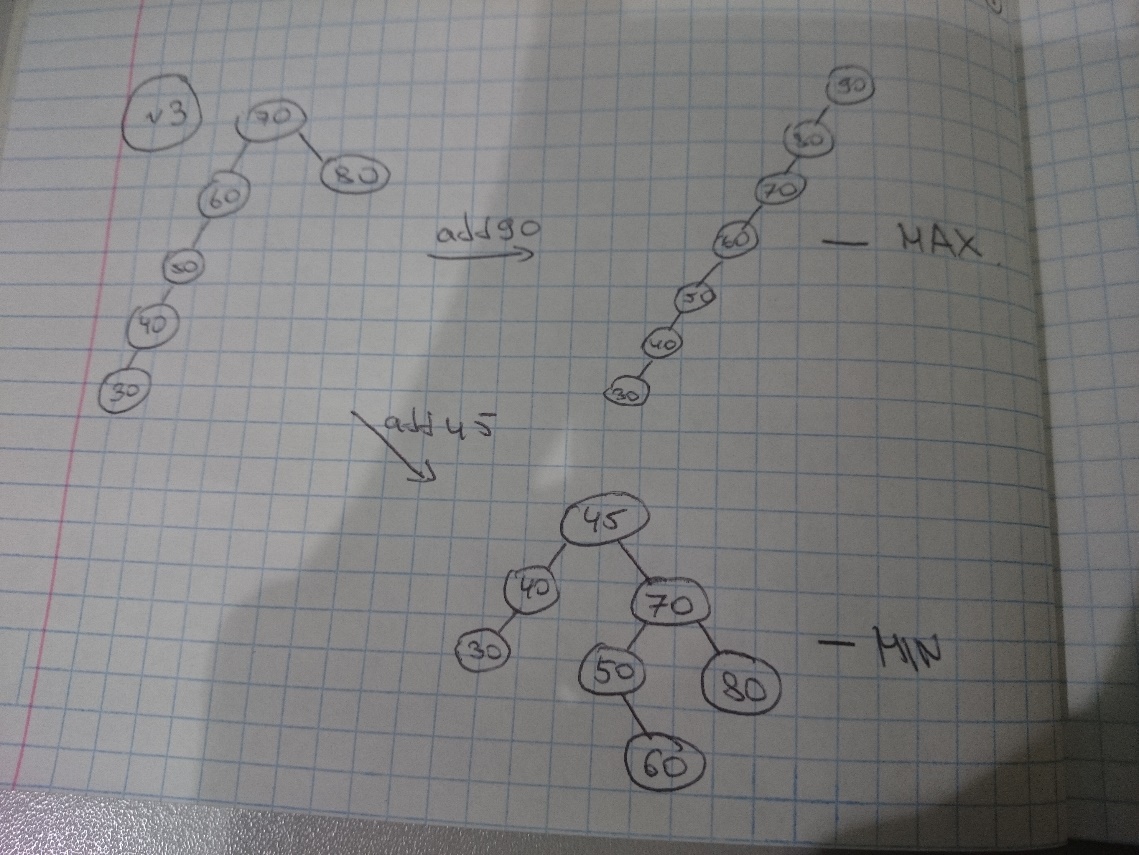
При zig – если есть альфа – все уже плохо. Поэтому у v должен быть b, но тогда у r нет правого сына.



Рассматривая крайние случаями, рассмотренными выше можно построить дерево из 6 вершин(один зиг зиг и один зиг). Чтд

***Задание 3***

Для начала построим дерево которое получится при добавлении всех вершин. Потом нам надо рассмотреть случаи при добавлении новой вершины. Очевидно, что максимальная глубина после очередного добавления будет если мы добавим большее число – ну на картинке это можно увидеть  
Минимальная глубина может быть если будет полное двоичное дерево. Так как вершина добавленная становится корнем, то очевидно, что это было число по середине. При добавлении 55 получаем, что дерево будет высоты 4 => высота как минимум 4. Ну я нарисовал пример для 45 – там тоже будет высота 4.



***Задание 4***

На рисунке показаны все стадии добавления каждых вершин

