Форма 4: разбор

1:

> Для B-дерева, как мы выяснили, важен выбор B. В базах данных его делают большим из-за особенностей работы с памятью. Давайте представим, что у нас таких особенностей нет. Мы просто делаем спуск по дереву глубины $\log_b(n)$ и в каждой вершине находим за b того сына, в которого хотим перейти. С помощью математического анализа можно найти оптимальное b (поскольку при b -> infty мы работаем за O(b), при $b = 2$ мы работаем за $log n$, а при $b = 1$ за n, то минимум будет в нуле производной, можно будет даже не проверять знаки). Попробуйте его найти.

> После этой операции рекомендую вспомнить 2-3 дерево и написать, почему мы взяли его, как пример, а не 3-5 или 5-9 дерево.

Мы знаем, что у B-дерева $log_b(n)$ уровней, сыновей b, а операции работают за высоту, умноженную на количество сыновей, то есть $b \log_b(n)$. Дальше можно посмотреть на эту картинку с вычислениями

Всё 2-3 дерево можно провязать двусвязными списками, связывающие вершины на одном уровне.

Почуму бы вместо 2-3 дерева не сделать бы 5-11 дерево? Оценим это так. Пусть у всех вершин степень d. Тогда стоимость операции:

$$d \cdot \log_d n = d \cdot \frac{\ln n}{\ln d} \Rightarrow$$

$$(d \cdot \log_d n)' = (d \cdot \frac{\ln n}{\ln d})' \Rightarrow$$

$$(d \cdot \log_d n)' = (d \cdot \frac{\ln d - 1}{(\ln d)^2})'$$

0 производной в $d = e$. Значит, 2 и 3 - то хорошие приближения.

2:

> При написании декартова дерева часто совершают ошибку и в merge делают не сравнение приоритетов, а бросок монетки: если рандом больше 0.5, то правый корень станет корнем объединения, иначе левый.

> Если в декартово дерево добавлялись элементы 1 2 3 4 5 6 .... n в таком порядке, найдите матожидание высоты такого дерева и напишите ее в ответ (желательно вместе с кратким описанием, как вы ее нашли). И после этого всегда сравнивайте случайные приоритеты, чтобы не напарываться на такую ошибку.

В этой задаче, поскольку числа добавляются по возрастанию, нам не обязательно делать split, и достаточно было посмотреть на самый первый if в мердже. А именно, на первом же сравнении, если монетка выпала орлом кверху, то мы подвесим правое дерево как корень, а левое сделаем его сыном. Таким образом, на каждом шаге высота дерева с вероятностью 1/2 увеличивается на 1, за счет чего получается совсем не логарифмическое матожидание n/2.