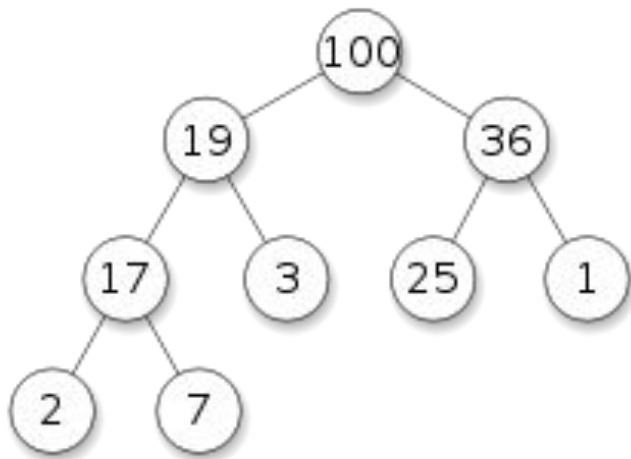
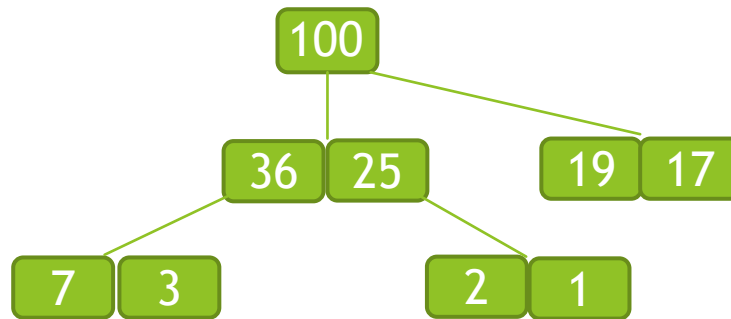


Вступление

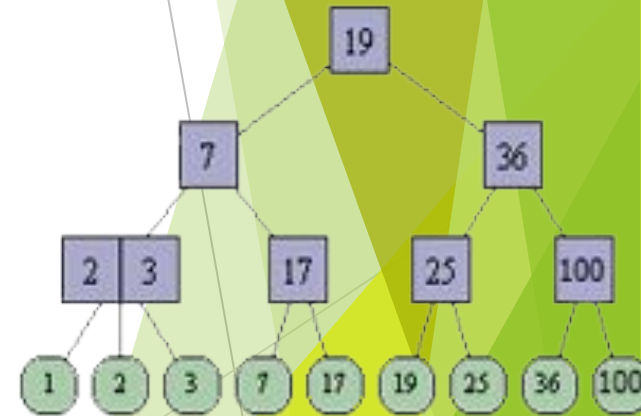
2-3 куча это массив 2-3 деревьев, обладающих свойствами куч(родитель больше(меньше) всех своих детей).
2-3 дерево это сбалансированное дерево, родительский узел которого может иметь как два, так и три сына.
2-3 кучи применяются для реализации очередей с приоритетом и являются оптимизацией Фибоначчиевых куч
Применяются для оптимизации алгоритмов на графах, работы с очередями, моделирования



Куча на примере бинарной

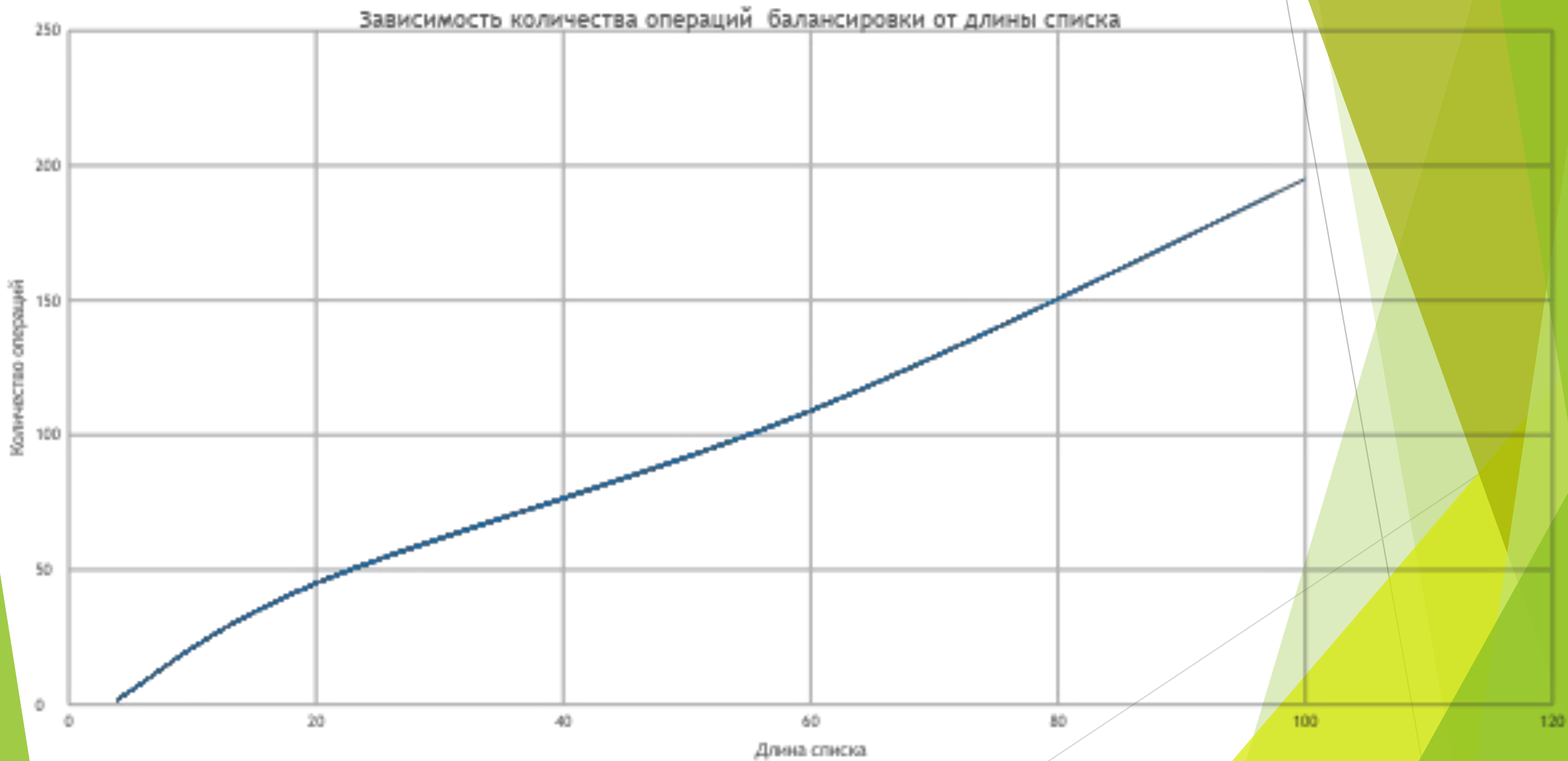


Одно из деревьев 2-3 кучи



2-3 дерево

Проблема Фибоначчиевых куч



Структура 2-3 кучи

Степень дерева - высота корневого узла

Насыщенность дерева - возможность добавлять в дерево вершины без увеличения его степени

Деревья бывают насыщенные(t) и ненасыщенные(f)

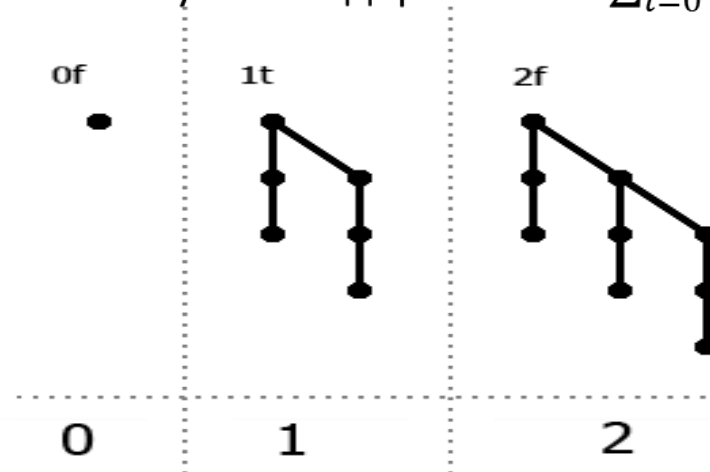
В куче хранится массив деревьев по следующему принципу:

В i -ой ячейке дерева располагается только дерево степени i

Количество элементов в ненасыщенном дереве = 3^n






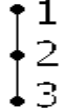

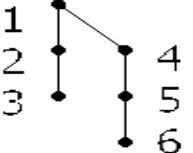
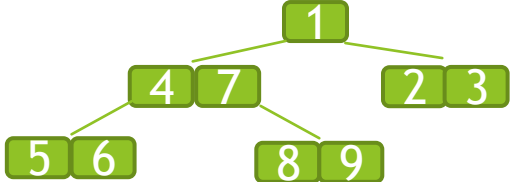
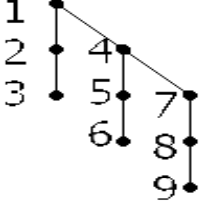
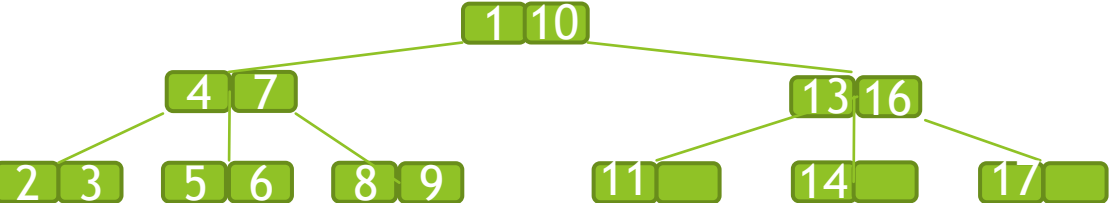
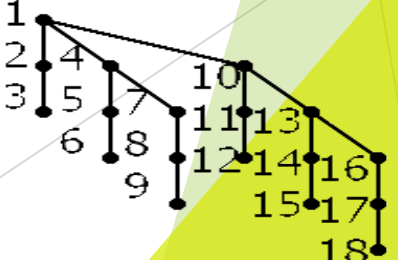
Количество элементов в насыщенном дереве = $2 * 3^n$

Максимальное количество элементов в куче из n деревьев = $\sum_{i=0}^n 2 * 3^i$



Куча из 16
элементов

Примеры деревьев

| Тип | Внешний вид 2-3 дерева | Общий внешний вид |
|-----|--|---|
| 0f |  |  |
| 0t |  |  |
| 1f |  |  |
| 1t |  |  |
| 2f |  |  |
| 2t |  |  |

Общая схема элемента

Для большей конкретности рассмотрим 1f
дерево:

K (key) - ключ, определяет приоритет элемента

V(value) - значение

pr(parent) - родительский узел, элемент с более
высоким

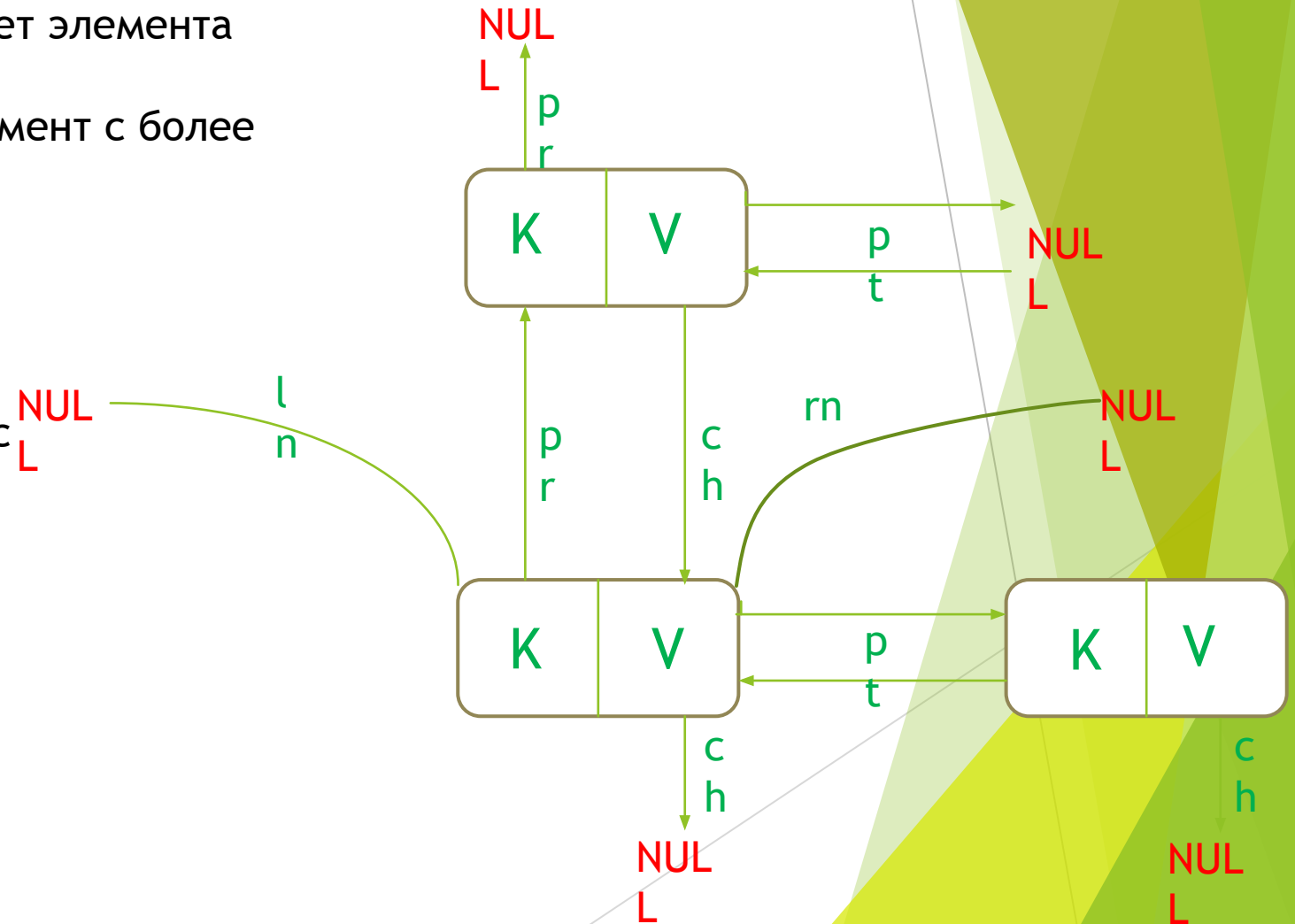
приоритетом.

pt(partner) - партнерский узел

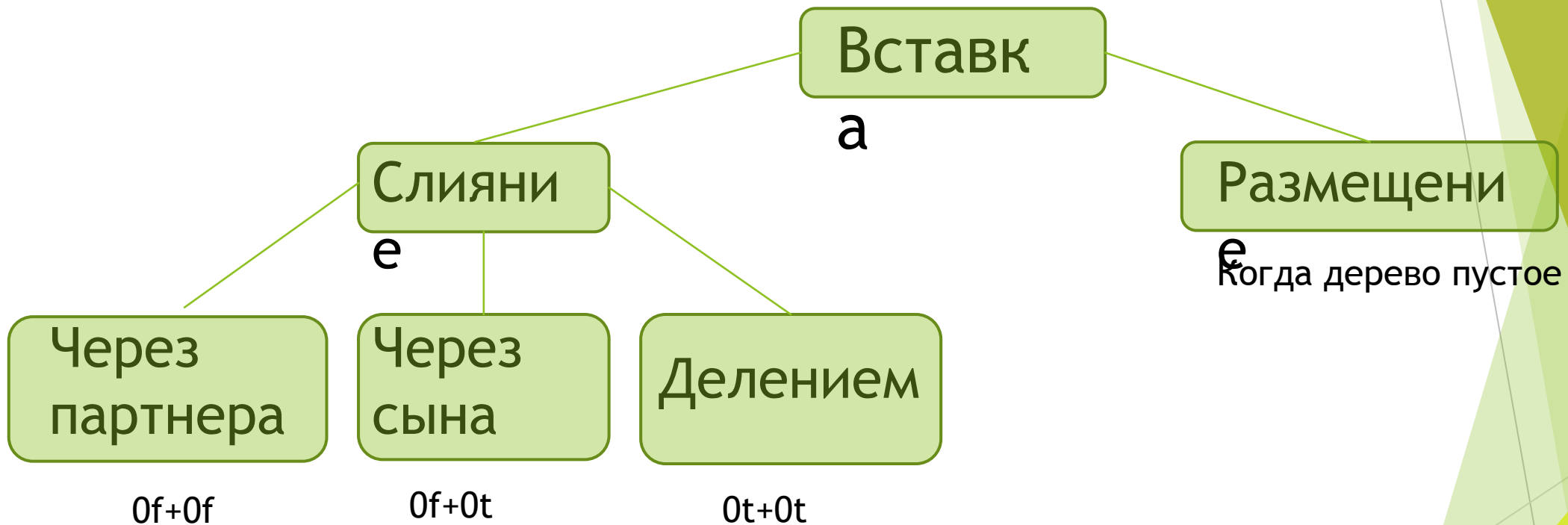
ln(left neighbor) - левый сосед

rn(right neighbor) - правый сосед

ch(child) - дочерний узел, элемент с
более низким приоритетом



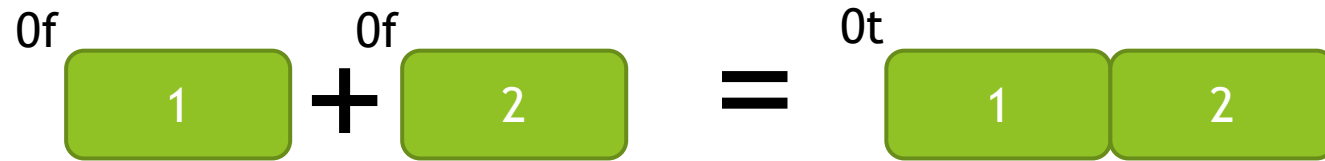
Вставка в кучу($O(1)$)



Пример вставки

1) Через партнера

$$nf + nf = nt$$



2) Через сына

$$nf + nt = (n+1)f$$



3) Делением

$$nt + nt = nf + (n+1)f$$



Извлечение из кучи минимума($O(\log N)$)

Для извлечения из кучи минимума(это корневая вершина) сначала найдем нужное дерево, а затем разложим дерево на составляющие по правилу:

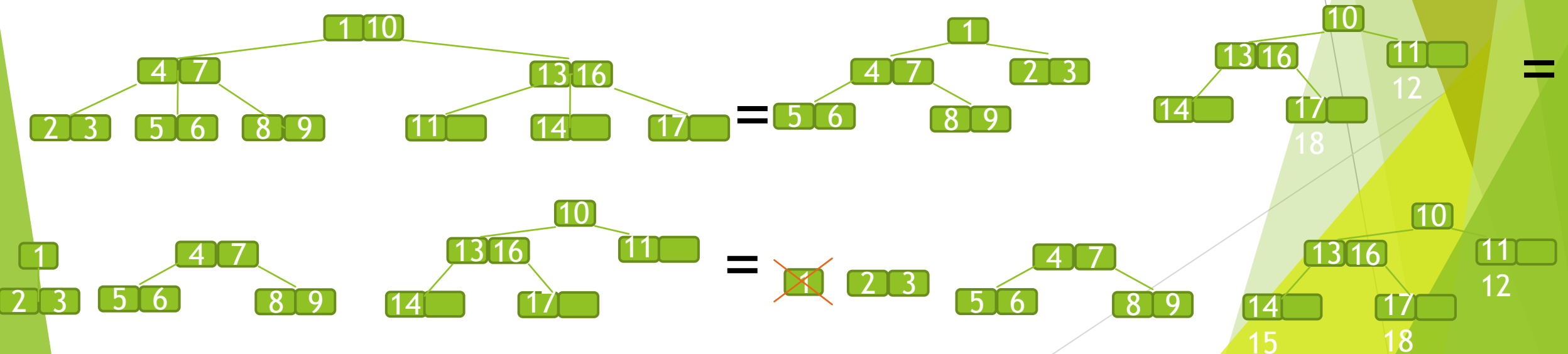
$$nf = (n-1)f + (n-1)t$$

$$nt = nf + nf$$

Раскладываем до тех пор пока не получим дерево из одной вершины. Далее вставляем части заново.

Пример:

$$2t = 2f + 2f = 1f + 1t + 2f = 0f + 0t + 1t + 2f$$

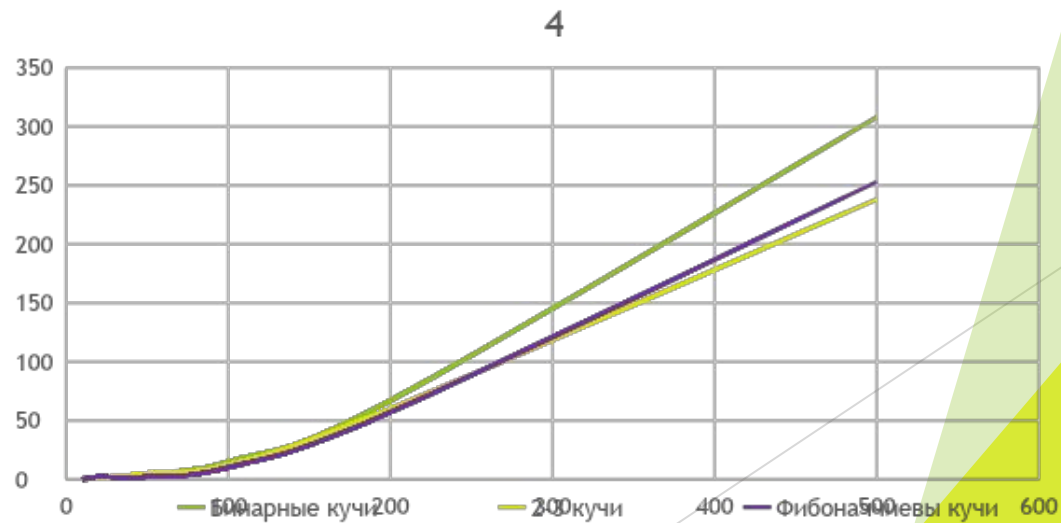
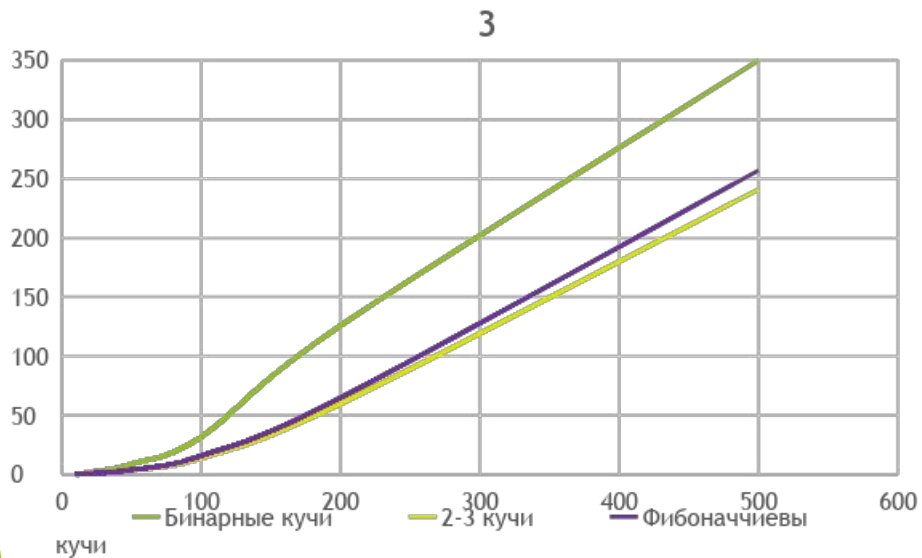
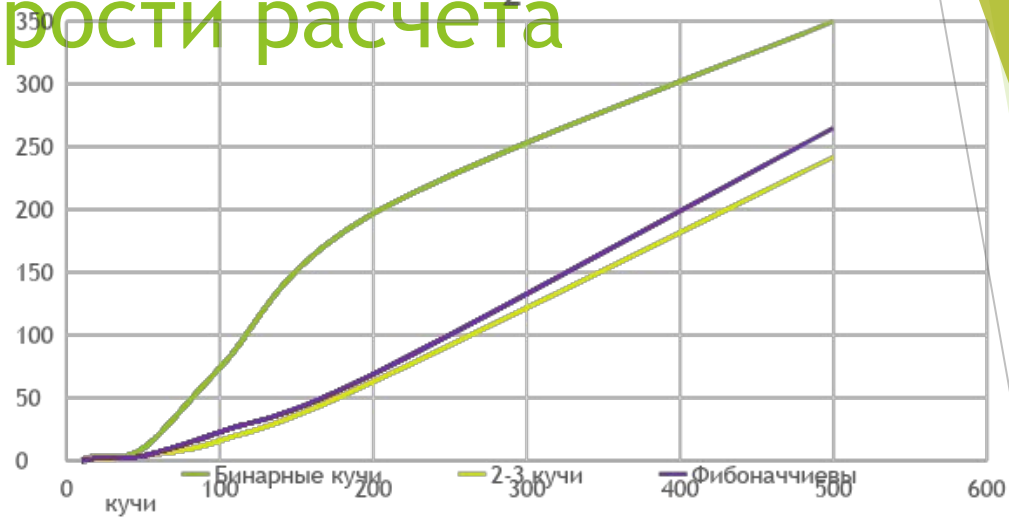
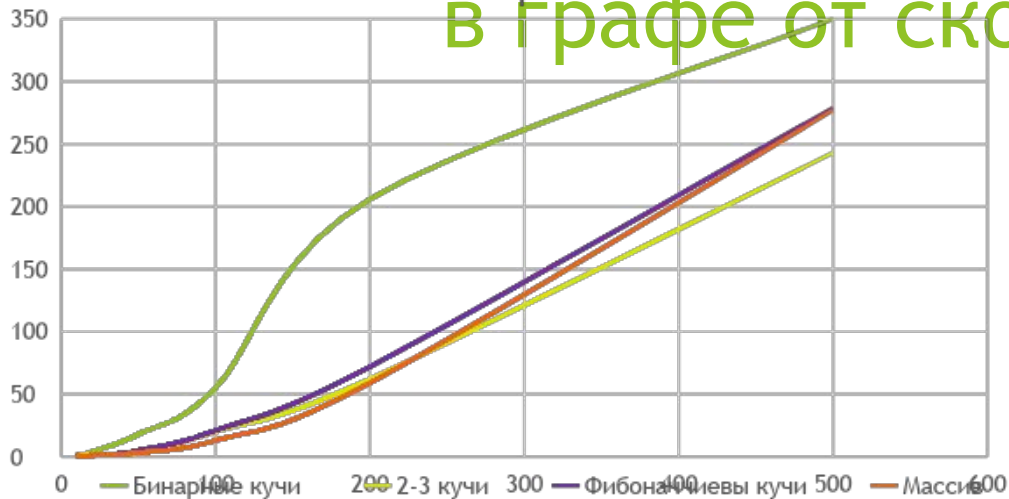


Сравнение асимптотик для различных куч

| Операции | Двоичная куча | Биномиальная куча | Фибоначчиева куча | 2-3 куча |
|---------------------|---------------|-------------------|-------------------|-------------|
| Вставка | $O(\log N)$ | $O(\log N)$ | $O(\log N)$ | $O(1)$ |
| Слияние | $O(N)$ | $O(\log N)$ | $O(1)$ | $O(1)$ |
| Поиск минимума | $O(1)$ | $O(\log N)$ | $O(1)$ | $O(1)$ |
| Извлечение минимума | $O(\log N)$ | $O(\log N)$ | $O(\log N)$ | $O(\log N)$ |
| Изменение ключа | $O(\log N)$ | $O(\log N)$ | $O(1)$ | $O(1)$ |

Зависимости количества вершин

в графе от скорости расчета²



Литература

1. C.A. Crane. Linear lists and priority queues as balanced binary trees. — Computer Science Dept, Stanford Univ. (1972).
2. Tadao Takaoka. [Theory of 2-3 Heaps](#). — Coccoon (1999).
3. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. — М.: МЦНМО, 2001. — С. 376-409.