



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Дальневосточный федеральный университет»  
(ДФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

---

Департамент математического и компьютерного моделирования  
ДОКЛАД  
о практическом задании по дисциплине АиСД

# Сбалансированные деревья: scapegoat tree

Выполнил студент гр. Б9121-09.03.03 пикд  
Козлова Светлана Евгеньевна  
Руководитель практики Доцент ИМКТ  
Кленин Александр Сергеевич

# Изобретение

1989 - Arne Andersson, Department of Computer Science, Lund University, Лунд, Швеция



LUND UNIVERSITY

1993 - Igal Galperin и Ronald Linn Rivest, Laboratory for Computer Science, Massachusetts Institute of Technology, Кеймбридж в штате Массачусетс, США



**Igal Galperin**

Laboratory for Computer Science, Massachusetts  
Institute of Technology, Cambridge, MA



**Ronald Linn Rivest**

Laboratory for Computer Science, Massachusetts  
Institute of Technology, Cambridge, MA

# Значение Scaregoat (англ. “Козел отпущения”)



В иудаизме особое животное, которое, после символического возложения на него грехов всего народа, обрекали на мучительную смерть.

Идиома используется в качестве метафоры и обозначает субъект, на который возложили ответственность за действия других, чтобы скрыть их настоящие причины.

# Scapegoat tree

Количество узлов  $q / 2 \leq n \leq q$

Высота  $\log \frac{1}{\alpha} q \leq \log \frac{1}{\alpha} 2n < \log \frac{1}{\alpha} n + 2$ ,  
 где  $\alpha$  — значение от  $1/2$  до  $1$

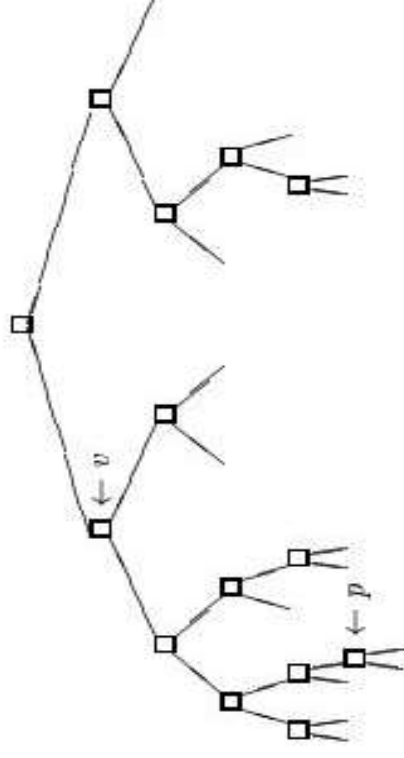
Проверка сбалансированности

$$1/2 \leq \alpha \leq 1$$

$$\text{size}(\text{left}[x]) \leq \alpha \cdot \text{size}(x);$$

$$\text{size}(\text{right}[x]) \leq \alpha \cdot \text{size}(x),$$

где  $\text{size}(\text{left}[x])$  и  $\text{size}(\text{right}[x])$  — размеры поддеревьев вершины  $x$



$p$  - вставленный элемент  
 $v$  - дисбаланс элемент,  
 требующий перестроения

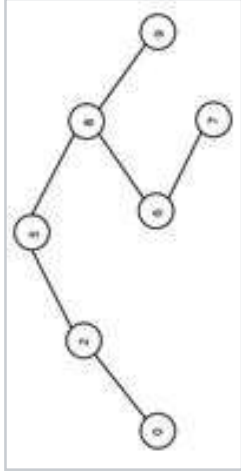
при коэф. дерева  $1,2$

# Достоинства

- Отсутствие необходимости хранить дополнительные данные в вершинах
- Отсутствие необходимости пересбалансировать дерево при операции поиска
- Амортизированная сложность операций вставки и удаления  $O(\log N)$
- Легкая модификация коэффициента «строгости» дерева  $\alpha$

# Недостатки

- В худшем случае операции модификации дерева могут занять  $O(n)$  времени
- Ошибочный коэффициент  $\alpha$  приведет к наихудшей производительности дерева



Пример дерева с  $\alpha = 0.6$

*Чем больше  $\alpha$ , тем глубже дерево,  
следовательно, запросы к дереву эффективные,  
а балансировка дерева не эффективная.*



# Заключение

Это первое двоичное дерево поиска, чьи операции в среднем составляют  $O(\log(n))$ , где  $n$  - количество узлов, и которое не занимает больше памяти, чем двоичное дерево поиска.

