

УДК 004.056:004.89

**РЕКУРСИВНЫЙ АЛГОРИТМ  
ФОРМИРОВАНИЯ  
ПСЕВДОСЛУЧАЙНОГО ДЕРЕВА***Галушка В. В., Мохаммад Джуман Хусейн*

Донской государственной технической  
университет, Ростов-на-Дону, Российская  
Федерация

[galushkavv@yandex.ru](mailto:galushkavv@yandex.ru)[eng.juman91@gmail.com](mailto:eng.juman91@gmail.com)

Рассматривается актуальная проблема представления иерархических структур данных в таблицах реляционных баз данных, возникающих из-за реальных практических потребностей при разработке сайтов. Предлагается автоматизированная система оценки эффективности различных методов представления иерархических структур данных в реляционных базах данных. В качестве основы для реализации предлагаемых средств разработан алгоритм генерации псевдослучайных деревьев для сокращения времени, затрачиваемого на формирование исходных данных в процессе тестирования. Описывается механизм работы этого алгоритма, отмечается его важность для упрощения работы предлагаемой системы и получения более достоверных результатов.

**Ключевые слова:** дерево, иерархия, база данных, структура данных, рекурсии.

**Введение.** Иерархическое представление данных является эффективным способом организации сложных информационных материалов и часто находит применение в современных информационных системах [1, 2]. Так, например, большинство веб-сайтов использует некоторую форму иерархической древовидной архитектуры либо в структуре самого сайта (деление на разделы, подразделы и т.д.), либо в представленной на нём информации [3]. При этом возникает задача представления такой структуры в реляционных базах данных. Для её решения существует множество способов, каждый из них имеет свои преимущества и характерные особенности, которые отличают его от других и определяют производительность и эффективность данного метода в конкретной ситуации [4–6]. Многообразие таких ситуаций и областей применения приводит к необходимости разработки автоматизированной системы оценки методов представления деревьев в реляционных базах данных, неотъемлемой частью которой является алгоритм генерации псевдослучайных деревьев, разрабатываемый с целью сокращения затрат времени на формирование исходных данных в процессе тестирования. Цель данной работы — проанализировать и описать методы и особенности реализации рекурсивного алгоритма генерации псевдослучайных деревьев для системы автоматизированной оценки методов представления иерархических структур в реляционных базах, которые позволяют упростить процедуру фор-

UDC 004.056:004.89

**RECURSIVE PSEUDORANDOM TREE  
GENERATION ALGORITHM***Galushka V. V., Mohammad J. H.*

Don State Technical University, Rostov-on-Don,  
Russian Federation

[galushkavv@yandex.ru](mailto:galushkavv@yandex.ru)[eng.juman91@gmail.com](mailto:eng.juman91@gmail.com)

The article deals with the actual problem of representing hierarchical data structures in relational database tables arising from actual practical needs in the development of sites. It offers an automated system for evaluating the effectiveness of various methods for representing hierarchical data structures in relational databases. As a basis for the implementation of the proposed tools, an algorithm for generating pseudorandom trees has been developed to reduce the time spent on the formation of the initial data in the testing process. The article describes the mechanism of operation of this algorithm and its importance for simplifying the work of the proposed system and for obtaining results that are more reliable.

**Keywords:** tree, hierarchy, database, data structure, recursion

мирования исходных данных и повысить достоверность получаемых результатов.

**Дерево как рекурсивная структура данных.** Деревья являются рекурсивной структурой данных и могут обрабатываться рекурсивными алгоритмами [7]. Рекурсивная структура состоит из более мелких структур такого же типа. Применительно к дереву это означает, что оно состоит из поддеревьев, каждое из которых также может включать в себя поддеревья, и т.д. Листовым узлом называется минимальное (размером один) поддерево. Рекурсивные алгоритмы работают таким же образом: часть функций выполняется путём вызова самого себя до тех пор, пока не будет достигнута наименьшая единица работы, для которой алгоритм может вернуть ответ [8]. Как только будет получен ответ от самого последнего уровня, управление передается обратно в копию функции, которая его вызвала, так что копия может завершить вычисления и так далее, пока вся цепочка вызовов не будет пройдена в обратном направлении и не вернется к первому вызову, который запустил её. В случае деревьев критерием, по которому определяется действие, выполняемое рекурсивной функцией, является наличие у узла потомков, лежащих на следующем уровне иерархии, вследствие чего листовая узел является условием остановки такой функции. Для остальных вершин их потомки рассматриваются как корневые узлы поддерева, каждое из которых обрабатывается отдельной копией рекурсивной функции.

**Рекурсивный алгоритм формирования дерева.** Рекурсия — фундаментальное понятие в математике и компьютерных науках. В языках программирования рекурсивной программой называется программа, которая обращается сама к себе (подобно тому, как в математике рекурсивная функция определяется через понятия самой этой функции). Рекурсивная программа не может вызывать себя до бесконечности, следовательно, вторая её важная особенность — наличие условия завершения. Таким образом, рекурсия в программировании может быть определена как сведение задачи к такой же задаче, но манипулирующей более простыми данными. Рекурсивный алгоритм всегда разбивает задачу на части, которые по своей структуре являются такими же, как исходная задача, но более простыми. Для решения подзадач функция вызывается рекурсивно, а их результаты каким-либо образом объединяются.

**Генерация псевдослучайного дерева.** После прояснения концепции рекурсии можно составить алгоритм, описывающий её применение для генерации деревьев. Исходными данными для него являются перечисленные ниже параметры:

— уровень — определяет количество вершин от текущего узла до корневого. Уровень корневого узла является нулевым и увеличивается после каждой рекурсии;

— ширина дерева — количество дочерних узлов, которое может быть создано для каждого родительского узла при каждом рекурсивном вызове. Ширина определяется как случайная величина, принимает свои значения в диапазоне [min\_width, max\_width];

— идентификатор родительского узла. Каждый раз, когда вызывается рекурсивный алгоритм, идентификатор узла, который выполняет этот вызов, передается в качестве родительского узла;

— контент — любой текст, который может быть использован как содержимое узла (категория, класс, группа и т.д.);

— глубина дерева представляет количество иерархических уровней в дереве, которое называется «Базовые критерии алгоритма», где рекурсивный вызов останавливается, когда количество иерархических уровней равно указанному иерархическому уровню (глубина);

— seed — ядро генерации случайных чисел, которое определяется функцией srand(seed). Важность его использования будет разъяснена ниже.

Далее приведены шаги по реализации алгоритма.

1. Первоначально алгоритм вызывается со следующими параметрами:

— уровень (level) = 0 (уровень корневого узла);

— исходный идентификатор корневого узла обрабатывается в соответствии с используемым методом и часто является NULL;

— ширина (width) дерева произвольно выбирается в пределах диапазона [min\_width,

min\_width];

— содержание (content): текст, который выражает содержимое узла.

2. Если уровень равен 0, корневой узел создается с помощью функции insert\_node, увеличивая уровень на единицу и вызывая алгоритм генерации дерева для текущего корневого узла (корневой узел становится родителем нового дерева).

3. Пока уровень меньше глубины (level < depth), создавать дочерние узлы в количестве, равном ширине (width), следующим образом:

— создать новый узел и внести его в базу данных;

— вызвать алгоритм генерации дерева для текущего узла.

4. Если уровень больше глубины (level > depth), то листовые узлы создаются в количестве, равном ширине (width).

Схема этого алгоритма показана на рис. 1.

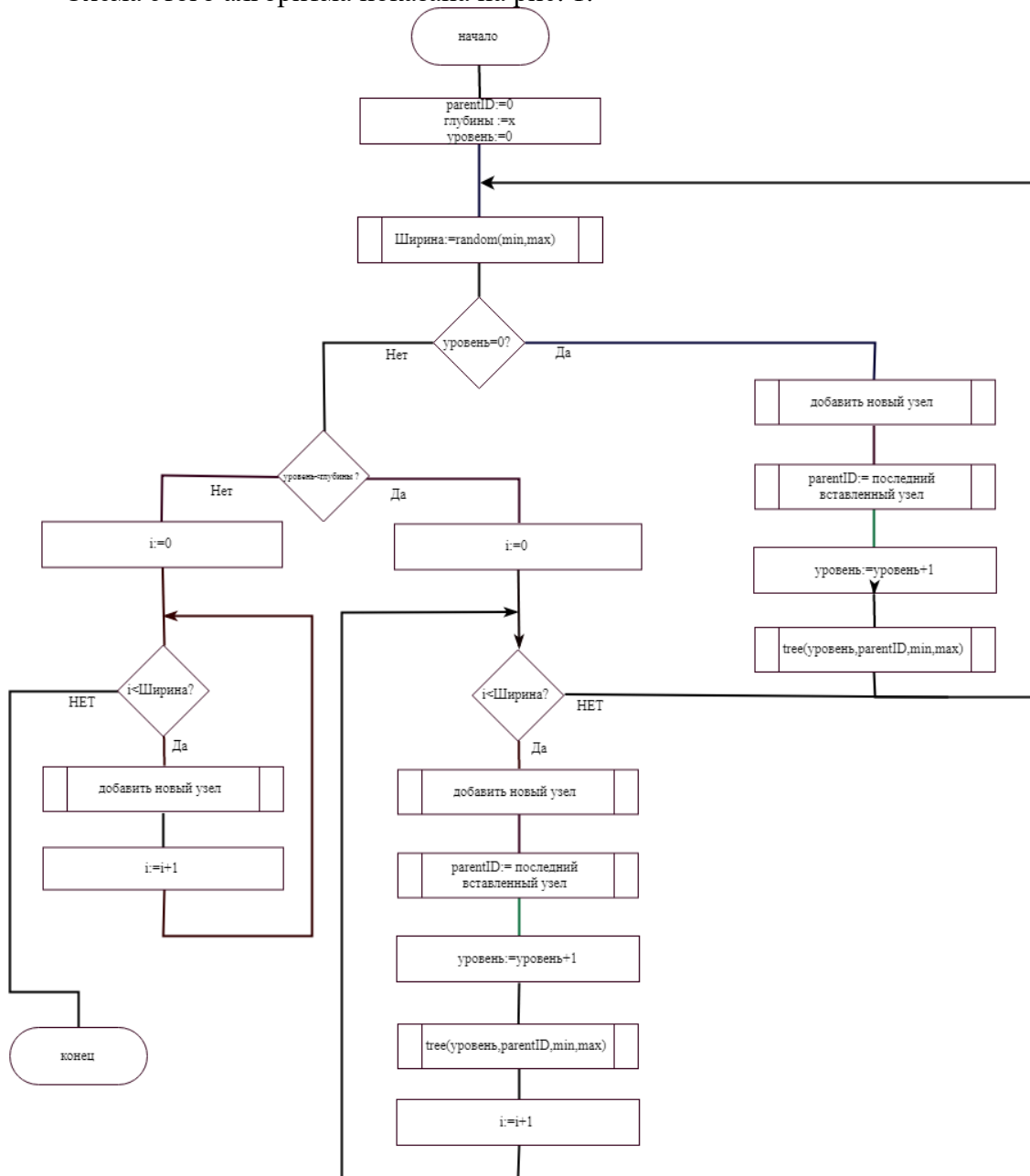


Рис. 1. Алгоритм генерации случайного дерева

**Особенности использования случайных чисел при создании дерева.** Как упоминалось выше, ширина дерева отражает количество потомков, которые генерируются для каждого узла, и является случайной величиной, принимающей значения между  $[\min\_width, \min\_width]$ . Наличие элемента случайности приводит к тому, что для унифицированных параметров деревьев (ширина и глубина) каждый раз при вызове описанного выше алгоритма получается другое дерево. Такой подход затрудняет сравнение эффективности методов представления деревьев в реляционных базах данных из-за невозможности получения двух одинаковых вариантов исходных данных. Для решения этой проблемы используется функция `srand (seed)`, которая изменяет начальное число генератора псевдослучайных чисел [9, 10]. Установка одинаковых значений `seed` приводит к тому, что случайные значения, генерируемые впоследствии для выбора основных параметров дерева, также будут одинаковыми, при этом они всё ещё будут отвечать требованиям равномерного распределения и непредсказуемости в рамках отдельного эксперимента. Таким образом, когда алгоритм вызывается с определенными параметрами глубины, ширины и начального значения генератора псевдослучайных чисел, он всегда производит одинаковые случайные деревья.

**Заключение.** Благодаря описанным выше особенностям реализации предлагаемый рекурсивный алгоритм генерации псевдослучайных деревьев для системы автоматизированной оценки методов представления иерархических структур в реляционных базах данных позволяет упростить процедуру формирования исходных данных и повысить достоверность получаемых результатов за счёт обеспечения повторяемости эксперимента.

#### Библиографический список

1. Новиков, Ф. А. Дискретная математика для программистов / Ф. А. Новиков. — Санкт-Петербург: Питер, 2000. — 304 с.
2. Мокшин, В. В. Рекурсивный алгоритм построения регрессионных моделей сложных вероятностных объектов / В. В. Мокшин, И. Р. Сайфудинов, А. П. Кирпичников // Вестник Казанского технологического университета. — 2017. — Т. 20, № 9. — С. 112–116.
3. Кирсанов, Д. Веб-дизайн: книга Дмитрия Кирсанова / Д. Кирсанов. — Санкт-Петербург : Символ-Плюс, 1999. — 376 с.
4. Тарасов, С. В. Способы реляционного моделирования иерархических структур данных / С. В. Тарасов, В. В. Бураков // Информационно-управляющие системы. — 2013. — № 6(67). — С. 58 — 66.
5. Древовидные (иерархические) структуры данных в реляционных базах данных [Электронный ресурс] / iBase.ru. — Режим доступа: <http://www.ibase.ru/treedb/> (дата обращения: 02.12.17).
6. Иванов, А. В. Представление и обработка иерархических данных в реляционных СУБД / А. В. Иванов, О. В. Зыбина // Математические методы в технике и технологиях — ММТТ. — 2015. — № 6. — С. 93–96.
7. Lennart L. Theory and practice of recursive identification / L. Lennart, T. Söderström // MIT press, 1983.
8. Celko J. Trees and hierarchies in SQL for smarties. — Elsevier, 2012.
9. Маклафлин, Б. PHP и MySQL. Исчерпывающее руководство / Б. Маклафлин. — Санкт-Петербург : Питер, 2013. — 512 с.
10. Малыхина, М. П. Базы данных: основы, проектирование, использование / М. П. Малыхина. — Санкт-Петербург: БХВ-Петербург, 2006. — 528 с.