На правах рукописи

Коротков Александр Евгеньевич

Методы, алгоритмы и программные средства повышения скорости поиска в базах данных

05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени кандидата технических наук

Автор: 

Москва – 2012

Работа выполнена в Национальном исследовательском ядерном университете «МИФИ».

Научный руководитель:	кандидат технических наук, доцент Кудрявцев Константин Яковлевич
Официальные оппоненты:	доктор технических наук, профессор, профессор кафедры «САПР» Московского государственного горного университета Петров Андрей Евгеньевич
	кандидат технических наук, доцент, доцент кафедры «Корпоративные информационные системы» Московского государственного технического университа радиотехники, электроники и автоматики Андрианова Елена Гельевна
Ведущая организация:	Федеральное государственное унитарное предприятие «Московский ордена Трудового Красного Знамени научно-исследовательский радиотехнический институт»

Защита состоится 28.12.2012 в 15 часов на заседании диссертационного совета Д 212.130.03 при Национальном исследовательском ядерном университете «МИФИ», расположенном по адресу: 115409, г. Москва, Каширское шоссе, 31.

С диссертацией можно ознакомиться в библиотеке университета.

Автореферат разослан 23.11.2012.

Отзывы и замечания по автореферату в двух экземплярах, заверенные печатью, просьба высылать по вышеуказанному адресу на имя ученого секретаря диссертационного совета.

Ученый секретарь

диссертационного совета Леонова Н. М.

Общая характеристика работы

Актуальность темы Информационные технологии интенсивно развиваются в современном мире и охватывают всё новые виды деятельности человека. В связи с этим накапливаются огромные массивы данных, и вопросы их обработки и хранения стоят особенно остро. При этом нагрузки на базы данных (БД) постоянно возрастают. Постоянно повышающиеся требования удовлетворяются как за счет совершенствования аппаратных средств, так и с помощью применения новых алгоритмов и структур данных. Таким образом, для современных БД, помимо современной аппаратной платформы, крайне важным является наличие высокопроизводительных методов доступа к данным.

Задачи повышения производительности носят разносторонний характер и требуют разработки различных алгоритмов, методов и структур данных при работе с геометрическими, текстовыми, мультимедийными и другими типами данных. Для повышения производительности обработки и извлечения данных в реляционных базах данных используются индексы и другие структуры данных.

Одной из наиболее широко применяемых структур данных является B-дерево. B-дерево обеспечивает высокую скорость выполнения запросов, связанных с типами данных и поисковыми предикатами, заданными SQL стандартом. Однако, для многих современных приложений необходима работа с типами данных и поисковыми предикатами, не заданными стандартом SQL. Примером таких типов данных могут служить геометрические типы данных (точки, многоугольники, кривые и т.д.), которые широко применяются в геоинформационных системах (ГИС). В ГИС также применяются не заданные SQL стандартом поисковые предикаты, такие как предикат пересечения геометрических объектов, предикат вхождения одного геометрического объекта в другой. Хотя для ГИС существуют различные стандарты, такие как Open GIS, структуры данных используемые СУБД для эффективной обработки поисковых запросов отличаются.

Существуют различные структуры данных, призванные обеспечить высокую скорость обработки поисковых запросов, касающихся пространственных данных, такие как разновидности grid-файлов, Quadtree и подобные ему деревья, R-дерево и подобные ему деревья. У каждой структуры данных есть свои преимущества и недостатки. R-дерево и его разновидности наиболее популярны по сравнению с другими перечисленными структурами данных, поскольку они могут эффективно работать во внешней памяти, а также могут хранить объекты, обладающие протяженностью, а не только точки. Основным недостатком R-дерева является то, что

охватывающие прямоугольники его узлов могут пересекаться. Сильная степень этого пересечения приводит к тому, что при поиске нужно сканировать много путей в дереве, что снижает скорость поиска. Способом уменьшения этого недостатка может быть новый алгоритм разделения узла для R-дерева, позволяющий уменьшить степень пересечения охватывающих прямоугольников узлов, тем самым увеличивая скорость поиска.

Наиболее распространенным типом данных является текстовый тип и задача повышения производительности поиска в текстовых массивах является одной из самых актуальных. В целом ряде практических задач таких как: очистка данных, смягчение запросов и проверка правописания не менее актуальным является нечеткий поиск в строковых массивах. При этом нечеткий поиск в СУБД не стандартизован, а самих мер похожести строк существует несколько. Среди различных мер похожести расстояние Левенштейна используется наиболее широко, поскольку оно применимо ко многим случаям. В настоящее время можно выделить две основные трудности при применении расстояния Левенштейна.

- Высокая вычислительная сложность расстояния Левенштейна при большом объеме исходных строк.
- Ограничения методов индексации для поиска по расстоянию Левенштейна в строковых массивах.

При этом для решения большинства практических задач, вычисление точного значения расстояния Левенштейна не обязательно, а достаточно его вычисления с пороговым значением.

Таким образом, вопросы повышения скорости поиска информации в базах данных являются актуальными и особенно для пространственных и текстовых данных.

Цель диссертационной работы состоит в повышении скорости поиска пространственных данных и разработке новых алгоритмов нечеткого поиска в массивах строк.

Для достижения поставленных целей необходимо решить следующие задачи.

1. Провести анализ алгоритмов построения R-дерева и его модификаций и разработать новый алгоритм разделения узла R-дерева, позволяющий улучшить качество результирующего R-дерева для пространственных данных.
2. Провести анализ существующих способов вычисления расстояния Левенштейна и разработать новый алгоритм вычисления расстояния Левенштейна с пороговым значением для поиска в текстовых данных.
3. Разработать индексную структуру на основе RD-дерева и k-грамм для поиска в наборах строк по расстоянию Левенштейна.

4. Реализовать разработанные алгоритмы, провести их экспериментальное исследование и внедрение.

Объектами исследований диссертационной работы являются базы данных, содержащие наборы пространственных данных и наборы строк.

Предметами исследований диссертационной работы являются алгоритмы и структуры данных, обеспечивающие высокую производительность при поиске многомерных данных и при нечетком поиске в наборах строк.

Методами исследования исследования диссертационной работы являются методы теории множеств, теории алгоритмов, линейной алгебры, математического анализа.

Научная новизна результатов работы заключается в следующем.

1. Разработан новый алгоритм разделения узла R-дерева для одномерного случая, основанный на использовании нового понятия «угловой разделяющей пары».
2. Разработано применение нового алгоритма разделения узла для R-дерева к многомерному случаю.
3. Разработан новый алгоритм вычисления расстояния Левенштейна с пороговым значением и математически доказана его корректность.
4. Разработана структура данных на основе RD-дерева и k-грамм для поиска в наборах строк по расстоянию Левенштейна.
5. Разработан алгоритм фильтрации сигнатур для структуры данных на основе RD-дерева и k-грамм, математически доказана его корректность.

Практическая значимость результатов диссертации заключается в следующем.

1. Новый алгоритм разделения узла R-дерева позволяет повысить скорость поиска пространственных данных.
2. Новый алгоритм вычисления расстояния Левенштейна с пороговым значением и предложенное применение RD-дерева к набору k-грамм индексируемой строки позволяют с большей скоростью решать задачи нечеткого поиска в наборах строк, что представляет практическую ценность при очистке данных, смягчении поисковых запросов и проверке правописания.

На защиту выносятся следующие основные результаты и положения:

1. новый алгоритм разделения узла R-дерева для одномерного случая, основанный на использовании нового понятия «угловой разделяющей пары»;

2. применение нового алгоритма разделения узла для R-дерева к многомерному случаю;
3. новый алгоритм вычисления расстояния Левенштейна с пороговым значением;
4. структура данных на основе RD-дерева и k-грамм для поиска в наборах строк по расстоянию Левенштейна;
5. алгоритм фильтрации сигнатур для структуры данных на основе RD-дерева и k-грамм.

Апробация работы Основные результаты диссертации докладывались на следующих конференциях:

- Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE) [1–3];
- IADIS International Conference Applied Computing [4];
- Международный научно-технический семинар в городе Алушта [5];
- Научная сессия МИФИ [6].

Реализация результатов диссертации заключается в следующем.

1. Программная реализация предложенного алгоритма разделения узла R-дерева была включена в исходных код наиболее развитой СУБД с открытым исходным кодом PostgreSQL.
2. Программная реализация предложенного алгоритма вычисления расстояния Левенштейна с пороговым значением была включена в исходных код наиболее развитой СУБД с открытым исходным кодом PostgreSQL.
3. Разработанный алгоритм разделения узла для R-дерева был применен в Государственном астрономическом институте им. П. К. Штернберга для поиска астрономических объектов.
4. Разработанный алгоритм разделения узла для R-дерева, а также методы нечеткого поиска текста были применены в ЗАО «Геноаналитика» для построения генетической карты и поиске формулировок заболеваний.

Публикации

Всего по теме диссертации опубликовано 12 печатных работ [1–12] в том числе 5 статьи в журналах, рекомендованных ВАК РФ для публикации основных результатов работы [7–11].

Личный вклад автора

Все научные и практические результаты диссертации получены автором лично.

Структура и объем диссертации Диссертация состоит из введения, четырех глав, заключения, списка использованной литературы из 115

наименований и одного приложения. Основная работа диссертации содержит 132 страницы текста, включая 47 рисунков и 7 таблиц.

Основное содержание работы

Во введении обоснована актуальность диссертационной работы, сформулирована цель и аргументирована научная новизна исследований, показана практическая значимость полученных результатов, представлены выносимые на защиту научные положения.

В первой главе проведен анализ основных алгоритмов и структур данных, служащих для поиска пространственных данных, а также для нечеткого поиска в наборах строк. Рассмотрены такие структуры для хранения пространственных данных, как Grid-файл, Quad-дерево, k-d-дерево, R-дерево, а также их разновидности. R-дерево и его разновидности наиболее распространенны для индексирования пространственных и других видов многомерных данных в силу следующих причин.

1. R-дерево позволяет индексировать как точки, так и объекты, обладающие протяженностью, в то время как многие структуры данных позволяют индексировать только точки.
2. R-дерево поддерживает различные виды запросов, такие как топологические запросы, запросы на поиск по направлению, запросы на поиск ближайших соседей.
3. R-дерево может эффективно работать во внешней памяти, в то время как многие структуры данных могут работать только в основной памяти, или их работа во внешней памяти затруднена.
4. R-дерево не накладывает дополнительных ограничений на индексируемые данные, таких как нахождение всех пространственных объектов внутри заранее заданных границ.
5. R-дерево может индексировать объекты в пространстве с произвольным числом измерений.
6. R-дерево обладает простой структурой, похожей на B-дерево, что позволило разработчиками легко встроить эту структуру данных в СУБД.

Было показано, что основным недостатком R-дерева является возможность пересечения минимальных охватывающих прямоугольников (МОП) его узлов. Степень пересечения МОП узлов зависит от исходных данных при построении дерева, а также от самой процедуры построения R-дерева, к которой на настоящий момент существует множество модификаций. Высокая степень пересечения МОП узлов, в конечном

счете, приводит к низкой скорости обработки поисковых запросов. В качестве средства смягчения этого основного недостатка R-дерева может выступать лучший алгоритм разделения узла. В связи с этим, актуальной задачей является разработка нового алгоритма разделения узла R-дерева, применение которого позволит сократить время выполнения поисковых запросов.

Также в первой глазе проведен анализ алгоритмов нечеткого поиска в строковых массивах. Одной из наиболее распространенных мер похожести строк является расстояние Левенштейна, поскольку данная мера применима к большинству задач. Расстояние Левенштейна – это минимальное число элементарных операций, необходимых для преобразования одной строки в другую. Элементарные операции могут быть следующими:

- вставка произвольного символа в произвольную позицию строки;
- замена произвольного символа строки другим произвольным символом;
- удаление произвольного символа строки.

Расстояние Левенштейна между строками a и b будем обозначать как $levenshtein(a, b)$. Для расчета расстояния Левенштейна между строками может быть использован алгоритм выравнивания двух последовательностей. Этот алгоритм основан на заполнении матрицы D размерами $n+1$ на $m+1$ (где n и m – длины строк a и b соответственно).

При решении таких задач, как: очистка данных, коррекция опечаток и смягчение поисковых запросов, необходимо найти в массиве строк те, которые похожи на поисковую строку. Таким образом, если в качестве метрики применяется расстояние Левенштейна, то задача состоит в том, чтобы из массива строк выбрать те, для которых это расстояние не велико, и именно для этих строк необходимо знать точное значение этого расстояния. Для тех строк, для которых значение расстояния Левенштейна велико, знать его точное значение не обязательно.

$$levenshtein\_threshold(a, b, k) = \begin{cases} levenshtein(a, b), levenshtein(a, b) \leq k \\ k + 1, levenshtein(a, b) > k \end{cases} = \\ = \min\{levenshtein(a, b), k + 1\} \quad (1)$$

Таким образом, была сформулирована задача нахождения расстояния Левенштейна с пороговым значением k. Если расстояние Левенштейна между строками a и b не превышает k, то необходимо узнать его точное значение. Если же расстояние Левенштейна превышает k, то достаточно только установления этого факта, точное значение расстояния знать

не обязательно. Эта задача может быть представлена как нахождение значения функции $levenshtein\_threshold(a, b, k)$ (формула 1) которая равна расстоянию Левенштейна между строками a и b, если оно не превосходит k, и равна $k + 1$ в противном случае. Задача нахождения значения функции $levenshtein\_threshold(a, b, k)$ может быть решена с временной сложностью $O(\min(n, m) \cdot k)$, за счет заполнения только части матрицы D вокруг её диагонали. Однако для многих задач, существующие алгоритмы расчета расстояния Левенштейна с пороговым значением всё ещё не достаточно быстры. Поэтому актуальной задачей является разработка более быстрого алгоритма расчета расстояния Левенштейна с пороговым значением.

Также в первой главе дается обзор метода разложения строки на k-граммы. K-грамма – это подстрока длины k, которая используется в качестве сигнатуры исходной строки. Число общих k-грамм само по себе может быть использовано в качестве меры похожести. Кроме того, число общих k-грамм может быть использовано для оценки других мер похожести.

Описана структура RD-дерева, а также алгоритмы его построения и поиска в нем. RD-дерево – это индексная структура для доступа к множествам. RD-дерево – это разновидность R-дерева. RD означает "Russian Doll" (матрешка), описывая тем самым фундаментальное свойство данной индексной структуры – рекурсивную вложенность. RD-дерево может применяться для ускорения различных запросов, оперирующих набором множеств.

Во второй главе описан новый алгоритм разделения узла R-дерева, разработанный в ходе данной диссертационной работы:

Разработанный алгоритм разделения одномерного R-дерева на основе двойной сортировки, позволяет более полно рассматривать разделения в одном измерении. Далее предложено расширение этого алгоритма на случай многомерного R-дерева, основанное на применении алгоритма одномерного разделения к каждой из осей.

В одномерном алгоритме разделения входные элементы содержат множество I интервалов x_i: $I = \{x_i\}$. Интервал x_i определяется своими верхней и нижней границами: $x_i = (l_i, u_i)$. Общая нижняя граница это $l = min\{l_i\}$, а общая верхняя граница – $u = max\{u_i\}$. Назовем пару <a, b> разделяющей парой, если любой интервал из I содержится либо в интервале (l, a), либо в интервале (b, u): $\forall x(x \in I \Rightarrow (x \subseteq (l, a)) \vee (x \subseteq (b, u))$. Будем называть разделяющую пару <a, b> угловой разделяющей парой, если $(a \in \{u_i\}) \wedge (b \in \{l_i\}) \wedge ((\forall t(t < a \Rightarrow \exists x(x \in I \Rightarrow (x \not\subseteq (l, t)) \wedge (x \not\subseteq (b, u)))) \vee (\forall t(t > b \Rightarrow \exists x(x \in I \Rightarrow (x \not\subseteq (l, a)) \wedge (x \not\subseteq (t, u))))))$.

На первом шаге алгоритма разделения узла одномерного R-дерева, осуществляется перечисление всех угловых разделяющих пар. Это делается на основе использования двух массивов: первый содержит входные элементы,

отсортированные по нижней границе, а второй содержит их же, но отсортированных по верхней границе. В главном цикле данного алгоритма осуществляется проход по двум массивам одновременно таким образом, что свойство разделяющей пары сохраняется.

Работа данного алгоритма проиллюстрирована на рисунке 1 для набора интервалов $[0, 5], [1, 7], [3, 6], [3, 8]$. Вначале находится первая угловая разделяющая пара <5, 0> (состояние 1 на рисунке 1). Далее осуществляется поиск следующего значения $s2.l$ и соответствующего $s1.u$. Этот шаг отображен как состояние 2 на рисунке 1, откуда видно что граница $s2.l$ увеличилась, но при этом не потребовалось увеличение $s1.u$. Далее осуществляется переход к следующему значению $s2.l$, таким образом находится угловая разделяющая пара <5, 1> (состояние 3 на рисунке 1). После этого осуществляется поиск следующего значения $s2.l$ и соответствующего $s1.u$. Далее находится промежуточная угловая разделяющая пара <6, 1> (состояние 4 на рисунке 1). После этого осуществляется переход к следующему значению $s2.l$, таким образом находится угловая разделяющая пара <7, 3> (состояние 5 на рисунке 1). Далее снова в осуществляется поиск следующего значения $s2.l$, при это оказывается что текущее значение уже наибольшее (состояние 6 на рисунке 1). И в завершение осуществляется переход к последней найденной угловой разделяющей паре <8, 3> (состояние 7 на рисунке 1).

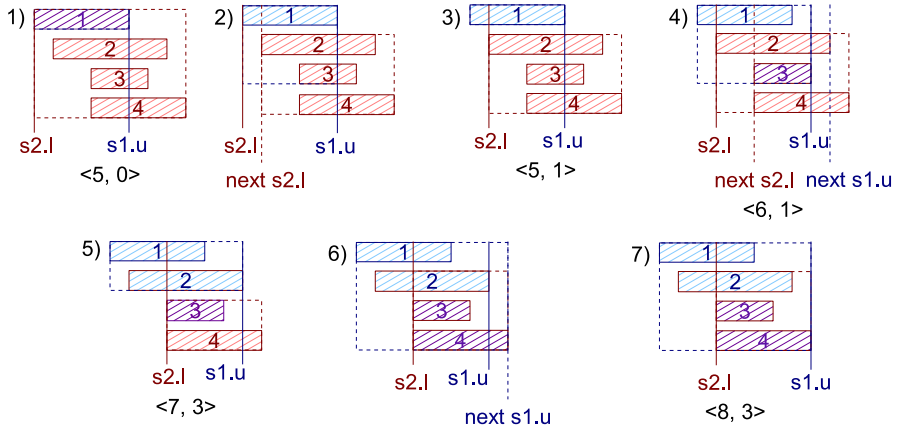

Рис. 1. Иллюстрация работы алгоритма перечисления угловых разделяющих пар

Среди всех перечисленных угловых разделяющих пар выбирается наилучшая по следующим критериям (перечислены в порядке убывания приоритета).

- Минимальное число элементов в группе больше или равно m.

- Степерь пересечения интервалов групп – наименьшая.
- Если охватывающие интервалы групп не пересекаются, то расстояние между охватывающими интервалами групп – наибольшее.
- Распределение элементов по группам наиболее близко к равномерному.

Было разработано применение разработанного алгоритма к многомерному случаю. На первом шаге перечисляются все угловые разделяющие пары вдоль всех осей и выбирается угловая разделяющая пара и соответствующая ось, для которых степень пересечений охватывающих интервалов – наименьшая. Если два или более разделения имеют одинаковую степень пересечения, то выбирается та ось, длина охватывающего интервала вдоль которой больше. Это стратегия делает МОП групп наиболее близкими к квадратам, что помогает при поиске.

В третьей главе описаны методы ускорения нечеткого поиска в наборах строк, разработанные в ходе данной диссертационной работы:

- новый алгоритм вычисления расстояния Левенштейна с пороговым значением;
- индексную структуру на основе RD-дерева и k-грамм для поиска в наборах строк по расстоянию Левенштейна.

В работе предложен новый алгоритм вычисления расстояния Левенштейна с пороговым значением, основанный на следующих принципах.

1. Диапазон заполняемых ячеек в строке определяется динамически на основании значений предыдущей строки, таким образом, достигается более полное отсечение необязательных вычислений.

2. К исходной матрице, используемой для вычисления расстояния Левенштейна, добавляется дополнительная матрица, которая характеризует минимально возможное расстояние Левенштейна между суффиксами строк. При этом сохраняются необходимые свойства, для рекуррентного заполнения этой матрицы и отсечений необязательных вычислений в ней. За счет того, что значения в этой матрице оказываются больше, чем в исходной, объем отсекаемых вычислений тоже оказывается больше. Тем самым достигается более высокое быстродействие.

Пример исходной матрицы (D) приведен на рисунке 2. Суммы исходной и дополнительной матриц (D'') приведен на рисунке 3.

Введем матрицу D^* размерностью n на m, такую что $D^*_{i,j} = \min\{D''_{i,j}, k+1\}$, где k – пороговое значение. Таким образом, $D^*_{n,m}$ будет отражать расстояние Левенштейна с пороговым значением.

		к	а	р	н	а	в	а	л
	1	0	1	2	3	4	5	6	7
к	2	1	0	1	2	3	4	5	6
а	3	2	1	0	1	2	3	4	5
р	4	3	2	1	0	1	2	3	4
а	5	4	3	2	1	0	1	2	3
в	6	5	4	3	2	1	0	1	2
а	7	6	5	4	3	2	1	0	1
н	8	7	6	5	4	3	2	1	0

Рис. 2. Матрица D строк «караван» и «карнавал».

		к	а	р	н	а	в	а	л
	1	1	3	5	7	9	11	13	15
к	3	1	1	3	5	7	9	11	13
а	5	3	1	1	3	5	7	9	11
р	7	5	3	1	1	3	5	7	9
а	9	7	5	3	2	1	3	5	7
в	11	9	7	5	4	3	1	3	5
а	13	11	9	7	6	4	3	1	3
н	15	13	11	9	7	6	5	3	2

Рис. 3. Матрица D'' для строк «караван» и «карнавал»

$$D^*_{0,0} = \min\{G(\delta_{i,j}) \cdot c_d + G(-\delta_{i,j}) \cdot c_i, k+1\} \tag{2}$$

$$D^*_{i,0} = \min\{D^*_{i-1,0} + H(-\delta_{i,j} - 1) \cdot (c_i + c_d), k+1\}, i \geq 1 \tag{3}$$

$$D^*_{0,j} = \min\{D^*_{0,j-1} + H(\delta_{i,j} + 1) \cdot (c_i + c_d), k+1\}, j \geq 1 \tag{4}$$

$$D^*_{i,j} = \min \left\{ \begin{array}{l} D^*_{i-1,j-1} + c_r \cdot d(a_i, b_i) \\ D^*_{i-1,j} + H(-\delta_{i,j} - 1) \cdot (c_i + c_d) \\ D^*_{i,j-1} + H(\delta_{i,j} - 1) \cdot (c_i + c_d) \\ k+1 \end{array} \right\}, i \geq 1, j \geq 1 \tag{5}$$

В диссертационной работе показано, что для матрицы D^* будут выполняться свойства 2, 3, 4 и 5.

Разработанный алгоритм вычисления расстояния Левенштейна с пороговым значением заключается в построчном заполнении матрицы E по рекуррентным формулам, выведенным ранее для матрицы D^*. Для каждой строки i сохраняется интервал $[l_i, u_i]$ содержащий номера ячеек строки, чьи значения не превосходят k. Заполнение $i+1$ строки начинается с ячейки l_{i-1} и заканчивается, когда одновременно текущий номер ячейки в строке больше $u_i + 1$ и значение ячейки больше k (или будет достигнут конец строки). Пример матрицы E на момент завершения работы алгоритма показан на рисунке 4.

		к	а	р	н	а	в	а	л
	1	1	3						
к	3	1	1	3					
а		3	1	1	3				
р			3	1	1	3			
а				3	2	1	3		
в					3	3	1	3	
а							3	1	3
н								3	2

Рис. 4. Матрица E для строк «караван» и «карнавал»

Также в третьей главе описана разработанная индексная структура на основе RD-дерева и k-грамм для поиска в наборах строк по расстоянию Левенштейна. Пример такого RD-дерева приведен на рисунке 5. В его листовых узлах вместо наборов k-грамм хранятся исходные строки. Во внутренних узлах хранятся сигнатуры. Сигнатура представляет собой битовый вектор, число бит в котором меньше, чем мощность множества возможных k-грамм. Это означает, что одному биту может соответствовать больше, чем одна k-грамма. Бит при этом устанавливается, если присутствует хотя бы одна k-грамма из соответствующих данному биту. Таким образом, если некоторый бит сигнатуры не установлен то это означает, что во всем поддереве нет ни одной из соответствующих этому биту k-грамм. Таким образом, сигнатура представляет собой фильтр Блума, т.е. компактное представление множества, допускающее ложноположительное срабатывание, но не допускающее ложноотрицательное. Для отображения k-грамм на биты сигнатуры использовалась хэш-функция crc32. В таком представлении, RD-дерево, за исключением листьев, становится очень похожим на S-дерево. Пример такого дерева приведен на рисунке 5.

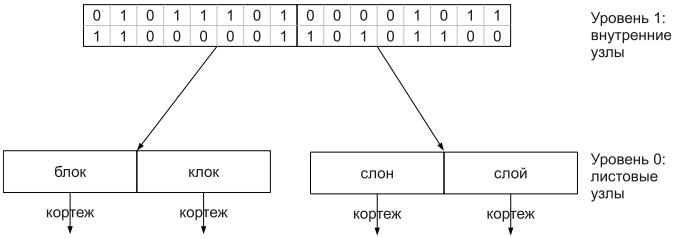

Рис. 5. Пример предлагаемого применения RD-дерева к наборам k-грамм

Для поиска по предложенному RD-дереву по расстоянию Левенштейна был разработан следующий алгоритм фильтрации сигнатур. Пусть $s1$ и $s2$ – некоторые строки. Если строку $s1$ разбить на n непересекающихся фрагментов и m из этих фрагментов не будет присутствовать в $s2$, то это означает что расстояние Левенштейна между строками $s1$ и $s2$ составляет не менее m. Если применить эту утверждение к k-граммам, то его можно сформулировать следующим образом. Если в строке $s1$ есть m непересекающихся k-грамм, не входящих в $s2$, то расстояние Левенштейна между строками $s1$ и $s2$ составляет не менее m. Применим это утверждение к фильтрации сигнатур. Если в поисковой строке s есть m непересекающихся k-грамм, для которых не установлен бит в сигнатуре, то минимальное расстояние Левенштейна между s и строкой поддерева – m. Таким образом, нужно найти наибольшее число непересекающихся k-грамм поисковой строки, для которых не установлен бит в сигнатуре. Работа этого алгоритма представлена на рисунке 6. Этот алгоритм последовательно рассматривает k-граммы строки s. Если бит в сигнатуре для рассматриваемой k-граммы не установлен, то счетчик k-грамм, которые нужно найти уменьшается на единицу, а следующие $k-1$ k-граммы пропускаются.

В четвертой главе проведена экспериментальная проверка производительности разработанных методов. Экспериментальная проверка разработанного алгоритма разделения узла R-дерева проводилось как на синтетических, так и на реальных наборах данных.

При экспериментальной проверке разработанного алгоритма разделения узла R-дерева использовались следующие реальные наборы данных:

- база данных Geonames, 7603617 точек (GN)[1];
- дороги Калифорнии, 2,249,727 МОП улиц Калифорнии (CAR)[2];

[1] `http://download.geonames.org/export/dump/allCountries.zip`

[2] `http://www.rtreeportal.org/datasets/spatial/US/CAR.tar.gz`

Рис. 6. Иллюстрация процесса фильтрации сигнатуры

- сеть каналов Tiger Streams, содержащая МОП 194,971 каналов штатов США Айова, Канзас, Миссури и Небраска (TS) [3].

В таблице 1 представлено среднее число доступов к узлам потребовавшееся для выполнения запроса для каждого набора данных, группы запросов и алгоритма разделения узла. Также в таблице 1 представлено среднее число строк, соответствующих каждой группе запросов каждого набора данных. Можно видеть, что разработанный алгоритм, основанный на двойной сортировке, превосходит все другие сравниваемые алгоритмы в отношении числа доступов к узлам при выполнении запроса (число доступов к узлам при использовании разработанного алгоритма меньше, что означает более высокую скорость поиска).

Для экспериментальной проверки скорости разработанного алгоритма расчета расстояния Левенштейна с пороговым значением использовались следующие реальные наборы данных:

- словарь английского языка объемом 98 тысяч слов, средняя длина слова $8,4$ знаков;
- словарь русского языка объемом 145 тысяч слов, средняя длина слова $11,3$ знаков;
- названия статей из списка DBLP. Всего $2,5$ миллиона названий, средняя длина названия – 47 знаков.

Результаты для названий статей из списка DBLP приведены в таблице 2. Экспериментальные результаты показывают значительное превосходство предложенного алгоритма по скорости работы, когда пороговое значение

[3] `http://www.rtreeportal.org/datasets/spatial/US/TS.tar.gz`

Таблица 1. Сравнение числа доступов к узлам на реальных данных (GQ – квадратичный алгоритм Гуттмана, NL – «новый линейный» алгоритм, R* – алгоритм разделения узла R*-дерева, DS – разработанный алгоритм)

Набор данных	Ср. число	Среднее число доступов к узлам			
		GQ	NL	R*	DS
GN	4,87	14,4 ± 0,5	219 ± 13	11,0 ± 0,2	7,3 ± 0,4
	11,07	16,9 ± 0,6	210 ± 14	12,1 ± 0,3	7,9 ± 0,5
	101,36	26,6 ± 1,3	263 ± 17	14,8 ± 0,3	10,2 ± 0,4
	998,70	52 ± 2	290 ± 20	29,8 ± 0,5	22,6 ± 0,7
CAR	1,32	7,5 ± 0,3	29,9 ± 1,4	7,0 ± 0,2	6,3 ± 0,2
	11,32	8,2 ± 0,2	31,9 ± 1,6	7,3 ± 0,2	7,1 ± 0,3
	102,93	11,4 ± 0,4	34,3 ± 1,6	10,3 ± 0,3	9,7 ± 0,3
	999,67	28,7 ± 0,5	62 ± 2	27,8 ± 0,5	26,1 ± 0,6
TS	1,00	4,87 ± 0,13	14,8 ± 0,5	4,30 ± 0,10	4,39 ± 0,10
	9,95	5,88 ± 0,16	16,6 ± 0,6	5,63 ± 0,13	5,21 ± 0,15
	99,92	9,0 ± 0,2	22,5 ± 0,7	8,65 ± 0,17	8,48 ± 0,19
	999,75	26,4 ± 0,3	46,2 ± 0,8	27,0 ± 0,3	25,3 ± 0,4

не велико (не превышает половины длины искомой строки). В этом случае скорость работы оказывается выше от 1,7 до 8 раз.

Для экспериментальной проверки предлагаемой индексной структуры на основе RD-дерева, примененного к набору k-грамм, использовались следующие реальные наборы данных:

- английский словарь из пакета Aspell в дистрибутиве Ubuntu linux;
- имена людей с Web сайта комиссии по переписи населения в штате Техас США.

Эксперименты проводились с различным значением k от 2 до 5 и различной длиной сигнатуры: 480 бит, 992 бит, 1984 бит и 1872 бит. В качестве поисковых строк выбирались случайные строки из набора данных. Значение порога выбиралось в интервале от 1 до 3.

Разработанная индексная структура сравнивалась с инвертированным деревом на k-граммах по размеру индекса и скорости поиска по индексу. Сравнение размеров индекса для словаря английского языка представлено на рисунке 8. Число доступов к узлам для поиска по словарю английского языка с пороговым значением $k = 2$ представлено на рисунке 7.

Таблица 2. Среднее время вычисления расстояния Левенштейна для базы данных DBLP (N – полное заполнение матрицы D, T – заполнение только части ячеек на основе их расположения, T – предлагаемый алгоритм)

Длина строки	Порог	Время вычисления, мс		
		N	T	P
1 – 10	1 – 10	5,79 ± 0,02	2,00 ± 0,02	1,11 ± 0,02
21 – 30	1 – 10	17,97 ± 0,10	5,20 ± 0,07	1,00 ± 0,02
	21 – 30	18,4 ± 0,2	13,0 ± 0,2	3,45 ± 0,11
41 – 50	1 – 10	27,70 ± 0,07	7,16 ± 0,07	1,18 ± 0,10
	21 – 30	28,02 ± 0,12	20,52 ± 0,12	4,63 ± 0,07
	41 – 50	28,87 ± 0,14	26,44 ± 0,15	12,30 ± 0,17
61 – 70	1 – 10	37,35 ± 0,08	9,26 ± 0,08	1,34 ± 0,10
	21 – 30	37,55 ± 0,10	26,16 ± 0,11	5,07 ± 0,06
	41 – 50	37,58 ± 0,10	34,36 ± 0,09	13,1 ± 0,10
	61 – 70	38,71 ± 0,14	38,82 ± 0,15	23,94 ± 0,18
81 – 90	1 – 10	47,71 ± 0,08	10,51 ± 0,12	1,37 ± 0,10
	21 – 30	47,50 ± 0,13	29,86 ± 0,18	4,17 ± 0,07
	41 – 50	47,39 ± 0,13	40,65 ± 0,13	11,56 ± 0,14
	61 – 70	47,73 ± 0,13	47,13 ± 0,12	23,21 ± 0,19
	81 – 90	48,5 ± 0,2	50,2 ± 0,2	36,7 ± 0,3
101 – 110	1 – 10	57,57 ± 0,13	10,88 ± 0,18	1,40 ± 0,02
	21 – 30	57,4 ± 0,3	32,9 ± 0,3	3,04 ± 0,08
	41 – 50	57,2 ± 0,2	45,0 ± 0,3	8,5 ± 0,2
	61 – 70	57,5 ± 0,3	53,5 ± 0,2	19,0 ± 0,4
	81 – 90	58,1 ± 0,3	58,9 ± 0,2	33,5 ± 0,4
	101 – 110	59,5 ± 0,4	62,0 ± 0,3	48,7 ± 0,8
121 – 130	1 – 10	68,1 ± 0,2	12,8 ± 0,3	1,50 ± 0,02
	21 – 30	68,2 ± 0,3	33,4 ± 0,5	2,05 ± 0,04
	41 – 50	68,1 ± 0,3	47,4 ± 0,4	4,67 ± 0,17
	61 – 70	67,9 ± 0,3	57,8 ± 0,3	12,25 ± 0,4
	81 – 90	68,0 ± 0,3	65,2 ± 0,3	26,19 ± 0,7
	101 – 110	67,9 ± 0,3	69,7 ± 0,3	44,7 ± 0,6
	121 – 130	68,3 ± 0,6	71,4 ± 0,6	59,47 ± 1,13
141 – 150	1 – 10	78,0 ± 0,4	11,7 ± 0,6	1,61 ± 0,02
	21 – 30	75,0 ± 0,3	35 ± 2	1,90 ± 0,09
	41 – 50	76,6 ± 0,7	48,1 ± 1,4	2,9 ± 0,2
	61 – 70	77,0 ± 0,8	59,8 ± 0,8	7,1 ± 0,6
	81 – 90	78,2 ± 0,8	69,7 ± 0,6	16,98 ± 1,16
	101 – 110	78,3 ± 0,7	76,4 ± 0,6	34,6 ± 1,8
	121 – 130	79,4 ± 0,7	81,5 ± 0,6	53,2 ± 1,6
	141 – 150	82,0 ± 1,4	85,5 ± 1,4	69,2 ± 3
161 – 170	1 – 10	91,1 ± 0,4	16,2 ± 0,7	1,75 ± 0,02
	21 – 30	91,8 ± 0,3	35,7 ± 0,7	1,79 ± 0,02
	41 – 50	91,4 ± 0,4	49,3 ± 0,6	2,03 ± 0,02
	61 – 70	91,0 ± 0,4	62,1 ± 0,5	3,33 ± 0,12
	81 – 90	90,9 ± 0,4	72,7 ± 0,5	7,7 ± 0,4
	101 – 110	90,5 ± 0,5	81,6 ± 0,5	19,7 ± 0,9
	121 – 130	91,4 ± 0,5	88,8 ± 0,3	39,0 ± 1,6
	141 – 150	90,5 ± 0,5	93,1 ± 0,4	64,9 ± 1,3
	161 – 170	90,6 ± 0,5	94,9 ± 0,5	82,6 ± 1,4

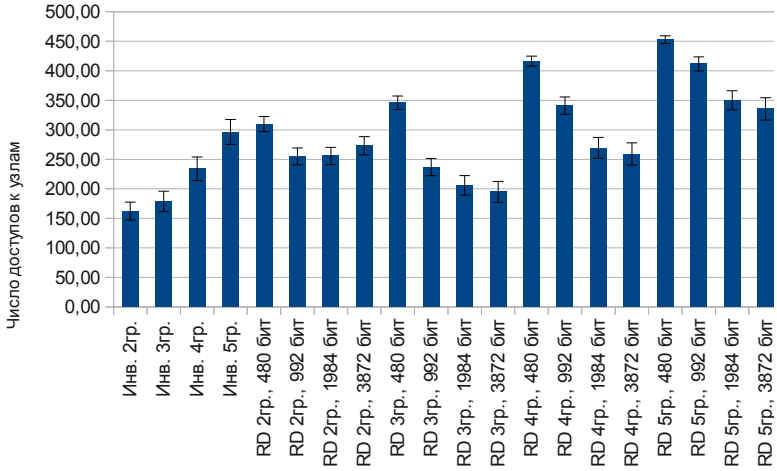

Рис. 7. Среднее число доступов к узлам при поиске по расстоянию Левенштейна с порогом 2 на наборе данных №1

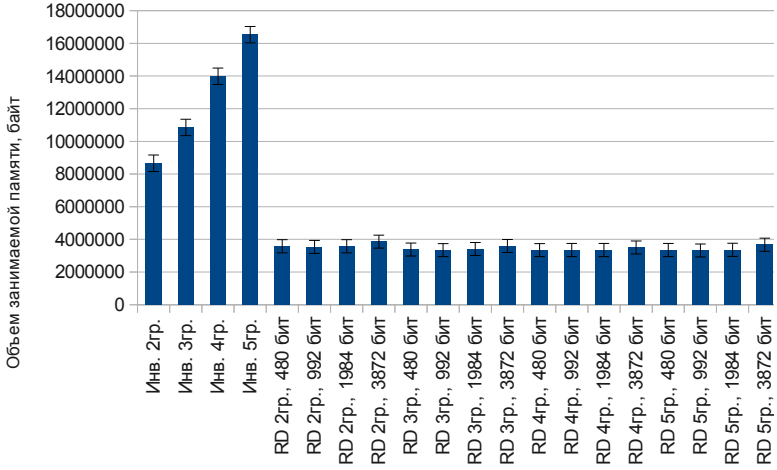

Рис. 8. Размеры индексных структур на наборе данных №1

Из рисунков видно, что при правильном выборе параметров RD-дерево на k-граммах способно обеспечить сравнимую скорость поиска при небольших пороговых значениях, при этом обладая существенно меньшим размером (в 2,3 – 7 раз меньше).

В заключении отражены основные результаты, полученные в данной диссертационной работе.

В приложение вынесены акты о внедрениях, а также исходные коды, разработанные в ходе данной диссертационной работы и включенные в код наиболее развитой СУБД с открытым исходным кодом PostgreSQL.

Основные результаты работы

1. Проведен анализ современных методов доступа к пространственным данным, а также методов нечеткого поиска в наборах строк. Сформулированы актуальные задачи.
2. Для разделения узла одномерного R-дерева введены понятия разделяющей пары и угловой разделяющей пары. Разработан алгоритм разделения узла одномерного R-дерева на основе двойной сортировки, осуществляющий перечисление всех угловых разделяющих пар, и его применение к многомерному случаю. Проведен анализ работы алгоритма в одномерном случае для различных входных данных.
3. Разработан новый алгоритм вычисления расстояния Левенштейна с пороговым значением. Этот алгоритм позволяет более полно отсекать вычисления, которые не могут повлиять на конечный результат.
4. Предложено применение RD-дерева к набору k-грамм для поиска в строковых массивах по расстоянию Левенштейна. Разработан алгоритм фильтрации сигнатур, позволяющий пропускать при сканировании больше поддеревьев, чем при фильтрации на базе числа общих k-грамм.
5. Проведена экспериментальная проверка разработанного алгоритма разделения узла для R-дерева, которое показало, что он превосходит аналоги в отношении числа доступов к узлам при поиске в большинстве рассмотренных случаев, при этом имея незначительно отличающееся время построения дерева.
6. Проведена экспериментальная проверка разработанного алгоритма вычисления расстояния Левенштейна с пороговым значением, которое показало, что разработанный алгоритм позволяет существенно сократить время поиска (ускорение от 1,7 до 8 раз), когда пороговое значение не велико (не превышает половины длины искомой строки).
7. Проведена экспериментальная проверка разработанного применения RD-дерева к набору k-грамм, которое показало, что разработанная индексная структура имеет размер от 2,3 до 7 раз меньше по сравнению с инвертированным деревом, сохраняя при этом сравнимое время поиска по расстоянию Левенштейна.
8. Разработанный алгоритм разделения узла для R-дерева был применен в Государственном астрономическом институте им. П. К. Штернберга для поиска астрономических объектов.

9. Разработанный алгоритм разделения узла для R-дерева, а также методы нечеткого поиска текста были применены в ЗАО «Геноаналитика» для построения генетической карты и поиске формулировок заболеваний.
10. Программная реализация разработанного алгоритма разделения узла R-дерева была включена в наиболее развитую СУБД с открытым исходным кодом PostgreSQL [4].
11. Программная реализация разработанного вычисления расстояния Левенштейна с пороговым значением была включена в исходных код наиболее развитой СУБД с открытым исходным кодом PostgreSQL [5].

Основные публикации по теме диссертации

1. Korotkov A. Database index for approximate string matching // Proceedings of the 4th Spring/Summer Young Researchers' Colloquium on Software Engineering. SYRCoSE '10. 2010. Pp. 136–140.

2. Korotkov A. Information system user interfaces automatic creation. SYRCoSE '09. 2009. Pp. 132–134.

3. Korotkov A. A new double sorting-based node splitting algorithm for R-tree // Proceedings of the 5th Spring/Summer Young Researchers' Colloquium on Software Engineering. SYRCoSE '11. 2011. Pp. 36–41.

4. Korotkov A. Automatic Creation of User Interfaces for Information System // Proceedings of the IADIS International Conference Applied Computing. IADIS '09. 2009. Pp. 327–329.

5. Коротков А. Е. Индекс базы данных для нечеткого поиска строки // Труды XIX международного научно-технического семинара. МНТС '10. 2010. С. 208–209.

6. Коротков А. Е., Панферов В. В. Применение обобщенного дерева поиска для нечеткого поиска текста // Труды научной сессии МИФИ-2010. 2010. С. 174–176.

7. Korotkov A. A new double sorting-based node splitting algorithm for R-tree // Programming and Computer Software. 2012. Vol. 38. Pp. 109–118.

[4] `http://goo.gl/u6RLI`

[5] `http://goo.gl/WS9lf`

8. Shelenkov A., Korotkov A., Korotkov E. MMsat—a database of potential micro- and minisatellites // Gene. 2008. Vol. 409, no. 1-2. Pp. 53 – 60.

9. Коротков А. Е., Панферов В. В. Применение обобщенного дерева поиска для нечеткого поиска строки // Наука и образование. 2011. Т. 3.

10. Коротков А. Е. Алгоритм разделения одномерных интервалов на группы при индексировании // Естественные и технические науки. 2012. Т. 1. С. 312–316.

11. Коротков А. Е., Трифонова Е. Е. Алгоритм расчета расстояния Левенштейна с пороговым значением // Естественные и технические науки. 2012. Т. 1. С. 317–322.

12. Кудрявцев К. Я., Коротков А. Е. Методы повышения скорости поиска информации в базах данных. LAP Lambert Academic Publishing, 2012. 84 с.

Подписано в печать 20.11.2012. Формат 60x84 1/16.
Объем 1,25 п.л. Тираж 100 экз. Заказ №261
Типография НИЯУ МИФИ. 115409, Москва, Каширское шоссе, д. 31