Методы сокращения времени поиска в слабоструктурированных базах данных

Механико-математический факультет
МГУ имени М.В. Ломоносова
Горелов С.С.
volerog@gmail.com

План доклада

- Краткий обзор методов сокращения времени поиска в базах слабоструктурированных данных
  - Предметные области
  - Модели данных
  - Методы логической оптимизации запросов
- Методы сокращения времени поиска при помощи индексов, представляющих собой иерархии схем OEM-документов
  - Постановка задач
  - Эффективность индекса
  - Построение индексов
  - Оценки сложности

Предметные области

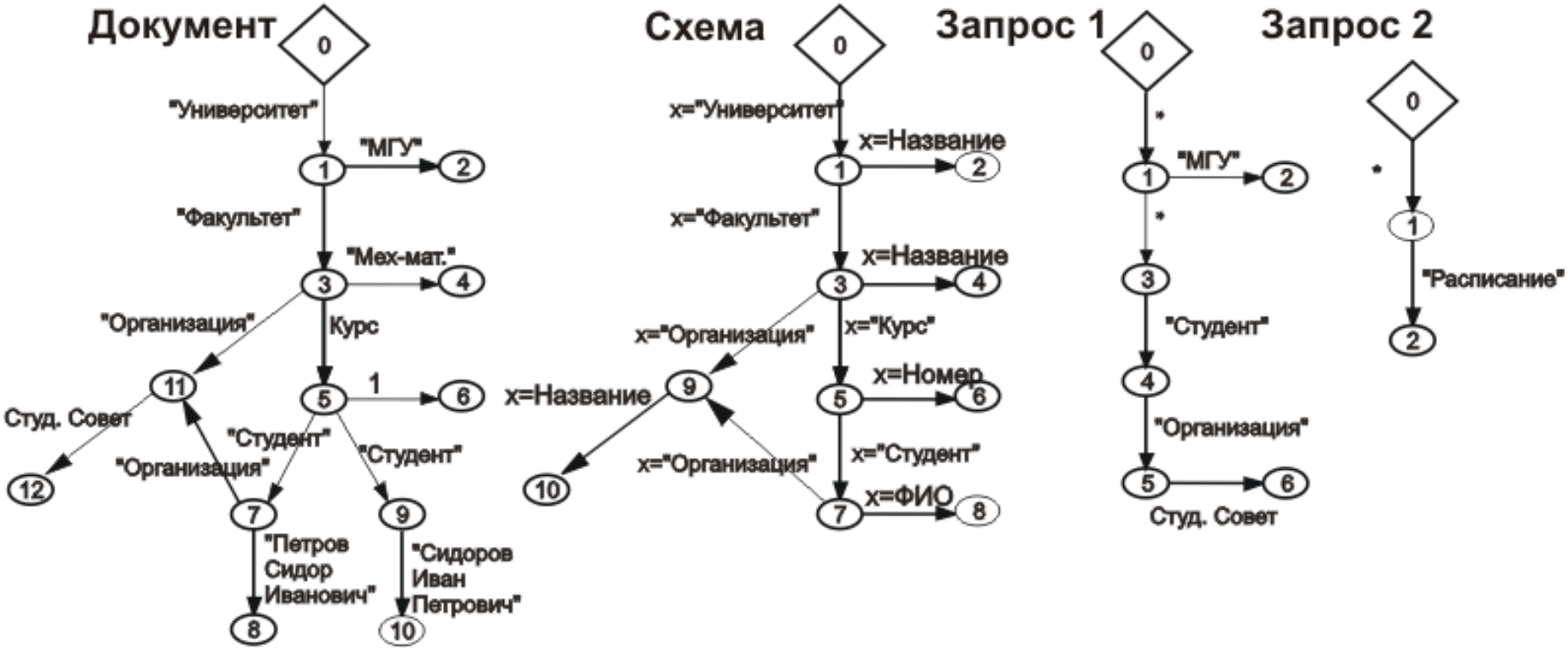
- Интеграция данных
- Semantic web
- Поиск в Интернет
- Специальные задачи в области биологии и химии

Модели данных

- XML-данные
  - Представление в виде дерева. Наличие атрибутов. Упорядоченность узлов
  - Обработка данных – XSLT, XQuery, XUpdate, XPath.
  - Схемы данных – XML Schema, DTD, Relax NG, структурные схемы.
- OEM-данные
  - Представление в виде графов.
  - Обработка данных – регулярные, конъюнктивные регулярные путевые запросы.
  - Схемы данных – графвые схемы.

Модель данных OEM

- **Документ** – ориентированный граф, ребра которого помечены символами алфавита.
- **База данных** – конечное множество документов.
- **Схема документа** – ориентированный граф, ребра которого помечены предикатами над алфавитом.
- **Запросы** – графы, ребра которых помечены регулярными языками над алфавитом.



Методы сокращения времени поиска

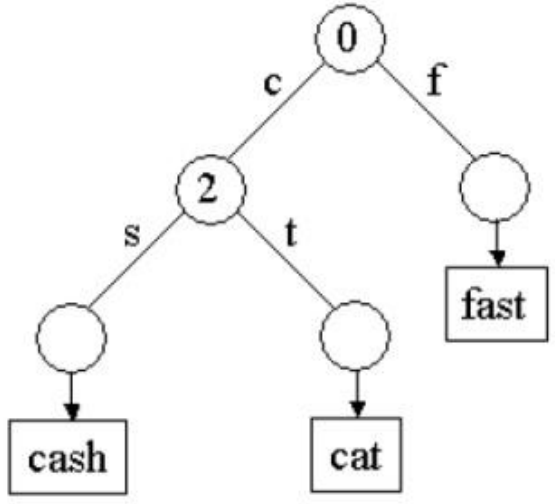
- Логическая оптимизация – основывается только на положениях модели данных.
- Физическая оптимизация – основывается на предположениях о способе хранения данных и выполнения запросов.

Логическая оптимизация

- Индексирование выражений.
- Полнотекстовые индексы.
- Индексирование путей.
- Индексирование подграфов.
- Перезапись запросов.
- **Усечение пространства поиска.**

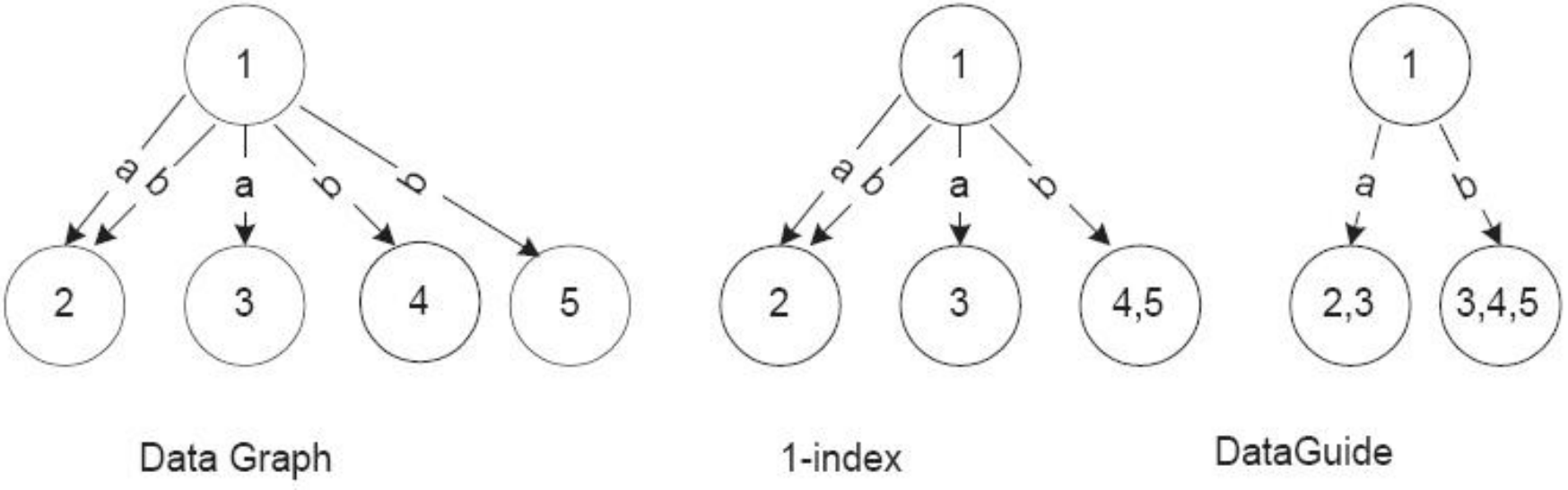
Индексирование путей. IndexFabric.

- Пути документа рассматриваются как строки
- Строки упорядочиваются в лексикографическом порядке
- Строки хранятся в виде дерева Patricia trie.



Индексирование подграфов

- Индекс – помеченный граф, для вершин которого задано некоторое отношение с вершинами документа.



Графовые индексы

- Dataguide – каждый путь меток в документе имеет не более одного образа в DataDuide, каждый путь в DataGuide имеет прообраз в документе.
- A(k)-индексы, D(k)-индексы, 1-индексы, T-индексы – отображения их вершин в вершины документа основаны на отношении похожести вершин графа.
- 1-индексы – вершины похожи, если все возможные пути от корня до вершин совпадают
- A(k)-индексы – вершины похожи, если все возможные исходящие пути длины <k совпадают
- D(k)-индексы – вводится понятие локальной похожести

Перезапись запросов с учетом схем данных

- Строится AND/OR граф запроса.
  - Каждое ребро запроса заменяется недетерминированным автоматом задающим регулярный язык, которым помечено ребро.
  - AND-вершины – терминальные и начальные вершины автоматов.
  - OR-вершины – промежуточные
- Строится декартово произведение схемы и AND/OR-графа, при этом остаются только ребра, для которых предикат с ребра схемы на метке запроса дает истину.
- Граф упрощается в соответствии с правилами о том, что AND-вершины должны быть достижимы из всех своих предков, а OR-вершины должны быть достижимы хотя бы из одного.

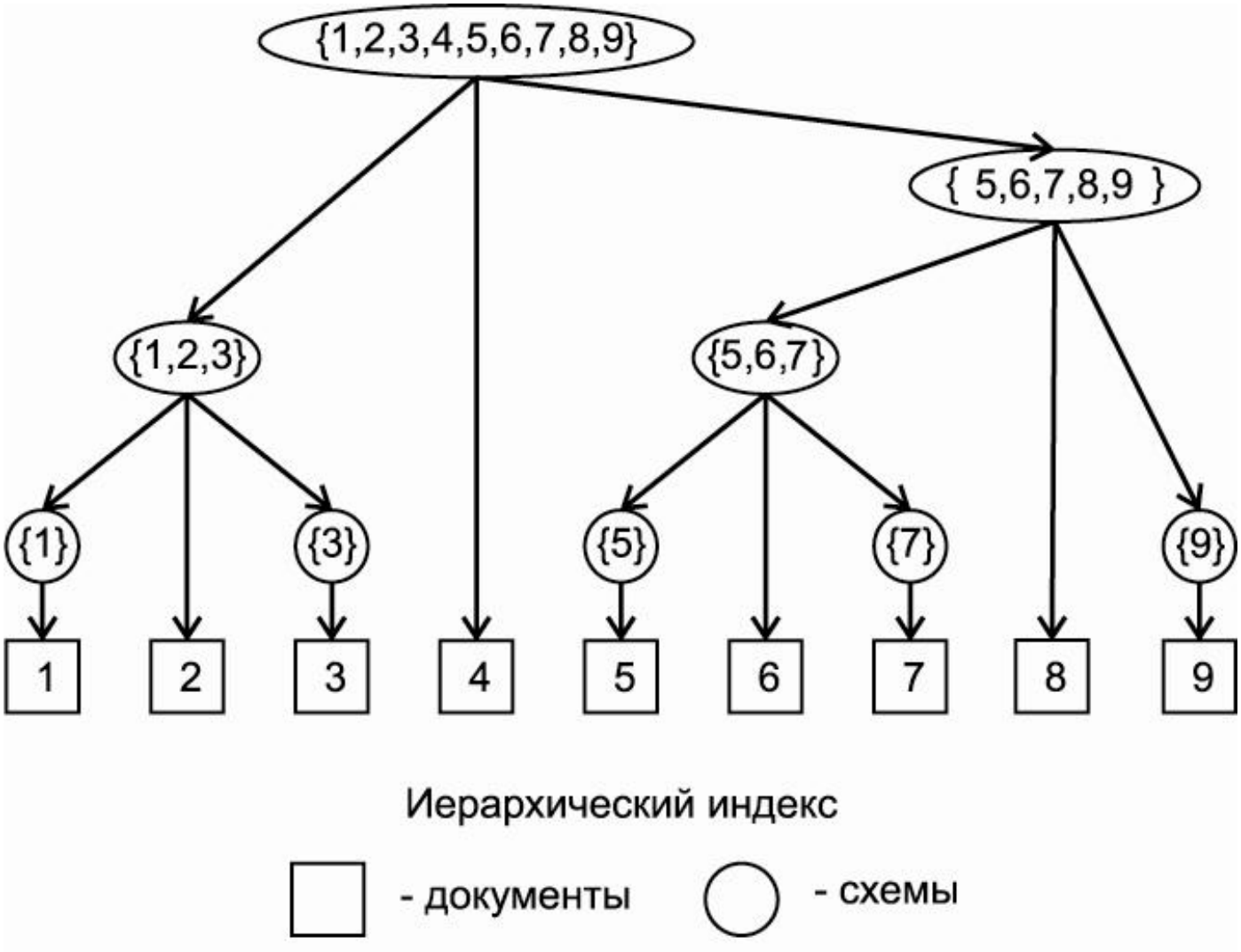
Свойства графовых схем

- Для схемы и документа можно вычислить отношение соответствия.
- Для схем можно вычислить отношение общности относительно множеств документов, которые им соответствуют.
- Для схемы и запроса существует алгоритм, показывающий в некоторых случаях, что по документам, соответствующим схеме не будет ничего найдено.

Усечение пространства поиска

- База данных – набор документов.
- Над документами задана иерархия схем.
- Алгоритм усечения пространства поиска
  - На каждом шаге алгоритма для рассматриваемой вершины индекса проверяем соответствие схемы и запроса.
  - Если условие не выполняется, то «отсекаем» рассматриваемую ветвь индекса со всеми соответствующими документами; если нет, то переходим к проверке дочерних схем.

Пример иерархии схем



Усечение пространства поиска для наборов однотипных документов. Задачи.

- Предложить формальный критерий сравнения иерархии в плане эффективности поиска.
- Предложить методы построения иерархии.
- Предложить методы добавления документов в иерархию.

Задача в общем случае. Основные понятия.

Будем полагать, что на рассматриваемом уровне абстракции, ни документы, ни запросы ни схемы не обладают какой-либо определенной структурой.

Определение

Документ — элемент наперед заданного множества **D**.

База данных *DB* — конечное множество документов.

Запрос — элемент наперед заданного множества **Q**.

Определение

Функция поиска документа — отображение $Qd : \boldsymbol{D} \times \mathbf{Q} \rightarrow \{0,1\}$.

Если $Qd(D, Q) = 1$, то это значит, что документ D соответствует запросу Q, и $Qd(D, Q) = 0$ в противном случае.

Определение

Схема S — объект, задающий множество документов (обозначим его [S]).

Схема S1 называется более общей чем S2, если $[S2] \subseteq [S1]$.

Также будем обозначать это отношение $S1 \geq S2$.

Индекс

Определение

Индексом (иерархией схем) I для базы данных *D* назовем такое дерево схем, что каждая схема является более общей, чем любая из ее дочерних схем.

Определение

Функция поиска по схеме — отображение $Qs : S \times Q \rightarrow \{0,1\}$. При этом $Qs(S, Q) = 1$, если существует такой D, соответствующий S, что $Qd(D,S)=1$ и $Qs(S, Q) = 0$, если такого D не существует.

Алгоритм, позволяющий сокращать время поиска в общем случае.

- На каждом шаге алгоритма для рассматриваемой вершины S индекса I проверяем соответствие схемы и запроса.
- Если условие не выполняется, то «отсекаем» рассматриваемую ветвь индекса со всеми соответствующими документами; если нет, то переходим к проверке дочерних схем.

Оптимальность индекса

Определение

Стоимость вычислений на схеме |S| - средняя сложность вычислений запросов по схеме S.

Определение

Вероятность схемы {S} - вероятность события, что Qs(Q,S)=1.

Математическое ожидание стоимости усечения пространства поиска по данному индексу / коротко назовем стоимостью индекса /.

Математическое ожидание стоимости равно:

$$M(I) = \sum_{S \in I} P\{\hat{S}\} * |S|.$$

Определение

Оптимальный индекс I_b - такой, что для любого другого индекса I

$$M(I) \geq M(I_b).$$

Построение оптимальных индексов по набору документов

Утверждение *Для произвольной вершины оптимального индекса ветвь, состоящая из всех её потомков, также является оптимальной иерархией.*

Основная идея:

Построение индекса необходимо осуществлять исходя из свойств локальной минимальности стоимости наилучшего индекса.

Общая схема алгоритма:

Алгоритм представляет собой итерационный процесс.

- На первом шаге процесса:
  - выделяем группу близких (в определенном смысле) схем,
  - строим схему обобщения и добавляем в индекс как родительскую
  - для объединяемых.
- На каждом следующем шаге рассматриваем верхний уровень схем и производим с ними описанные выше операции.

Построение оптимальных индексов по потоку документов

- Для документа строится схема и добавляется как дочерняя для какой-нибудь вершины уже существующего индекса.
- Вершина индекса в которую добавляется документ ищется из соображений минимизации стоимости полученной при модификации индекса.
- Части индекса, не допускающие документ, не имеют с ним ни чего общего и при добавлении документа не изменяются.
- При добавлении документов в одну и ту же ветвь индекса наступает момент времени, когда эту ветвь необходимо перестраивать.

Модель поиска.

Компоненты:

- **D,Q,S, вероятностное пространство**

Функции:

- построение схемы по документу - S(D): **D->S**;
- вычисление размера схемы - |S|: **S**->R;
- отношение на схемах - $S_1 > S_2$: **S**x**S->**{0,1};
- объединение схем - $S_1 + S_2$: **SxS->S**;
- вычисление вероятности схемы - P{S}: **S->R**;
- вычисление запроса на документе - Qd(D,Q): **DxQ**->{0,1};
- вычисление запроса на схеме - Qs(S,Q): **SxQ**->{0,1};
- проверка соответствия документа схеме – S>D: **SxD**->{0,1}.

Свойства:

- отношения $S_1 > S_2$, S>D и операция $S_1 + S_2$ соответствуют некоторому изоморфизму **S** и подмножества $2^{\mathbf{D}}$;
- если верно Qs(S,Q)=1, то существует $D \in S$: Qd(D,S)=1;
- если верно Qs(S,Q)=0, то для любого $D \in S$: Qd(D,S)=0.
- функции P{S} и |S| соответствуют своим определениям для вероятностного пространства.

Иерархии схем слабоструктурированных документов

Множества модели:

- **D**=множество OEM документов.
- **Q**=множество CRP запросов.
- **S**=множество графовых схем.

Описаны функции необходимые для того, чтобы задать модель поиска для описанного случая.

Предложен метод задания вероятностного пространства для случаев поиска в OEM по элементарным регулярным запросам и регулярным путевым запросам (CRP).

Для заданного вероятностного пространства введены операции отношения на схемах, вычисления их вероятности, сложения схем.
Доказаны требуемые для таких операций свойства.
Получены оценки сложности алгоритмов их вычисления.

Вероятностное пространство регулярных путевых запросов

Теорема 1. Для любого N существует M такое, что для любого документа D: |D|<N и произвольного регулярного запроса Q существует запрос Q’, состоящий из слов длины меньше M, при этом Q’=Q.

Теорема 2. Для любого N существует M такое, что юля любого документа D: |D|<N и произвольного CRP запроса Q, существует CRP запрос Q’, регулярные языки на ребрах которого содержат лишь слова длины меньше M и при этом Q=Q’.

Теорема 3. Для любого вероятностного пространства запросов существует наилучшая иерархия схем, такая, что стоимость любой другой иерархии не меньше, чем у наилучшей.

Приложения в области XML данных

Зададим множества модели:

- D = XML-документ.
- Q = элементарный XPath-запрос.
- S = DTD-схема.

Описаны функции необходимые для того, чтобы задать модель поиска для описанного случая.

Предложен метод задания вероятностного пространства для множества элементарных XPath-запросов.

Введены операции над DTD-схемами и показано выполнение свойств, необходимых для корректности модели поиска.

Оценки сложности алгоритмов.

Пусть база данных состоит из N документов, сложность вычисления |S| оценивается, как $C_{|S|}$, сложность вычисления $P_{\{S\}}$ как $C_{P\{S\}}$, а сложность вычисления S_1+S_1 как C_{S1+S2}.

Для разработанных алгоритмов доказаны следующие оценки сложности.

1. По набору документов: $O(NC_{|S|}+N^2(C_{S1+S2}+C_{P\{S\}})+N^3\ln(N))$.
2. По потоку документов: $O(N(C_{|S|}+C_{S1+S2}+C_{P\{S\}})+N^2\ln(N))$.

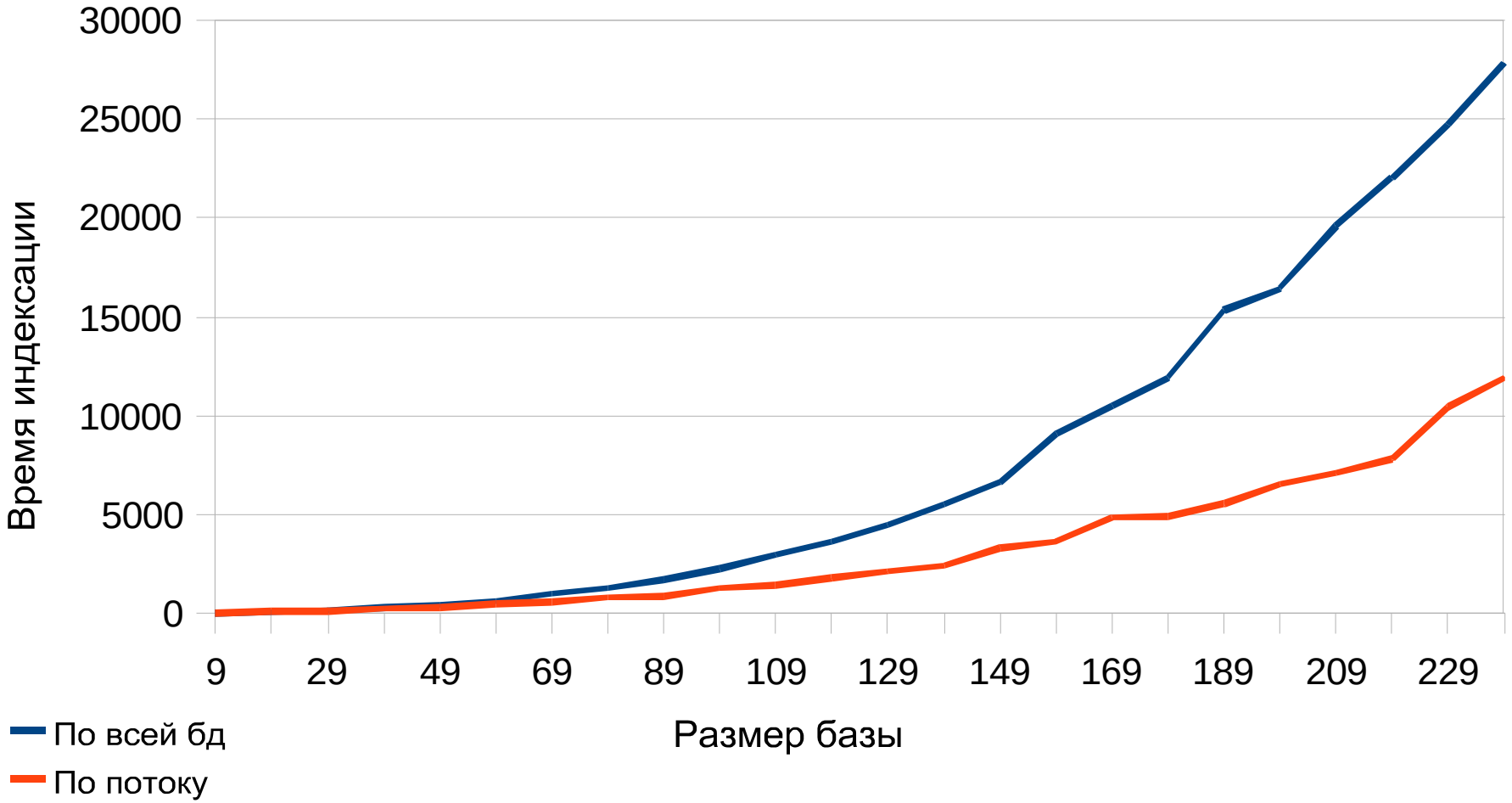
Оценки сложности отдельных операций для поиска в XML-документах по элементарным XPath-запросам:

$$C_{|S|}= O(|S|) \quad C_{S1+S2} = O(|S|) \quad C_{P\{S\}} = O(|S|)$$

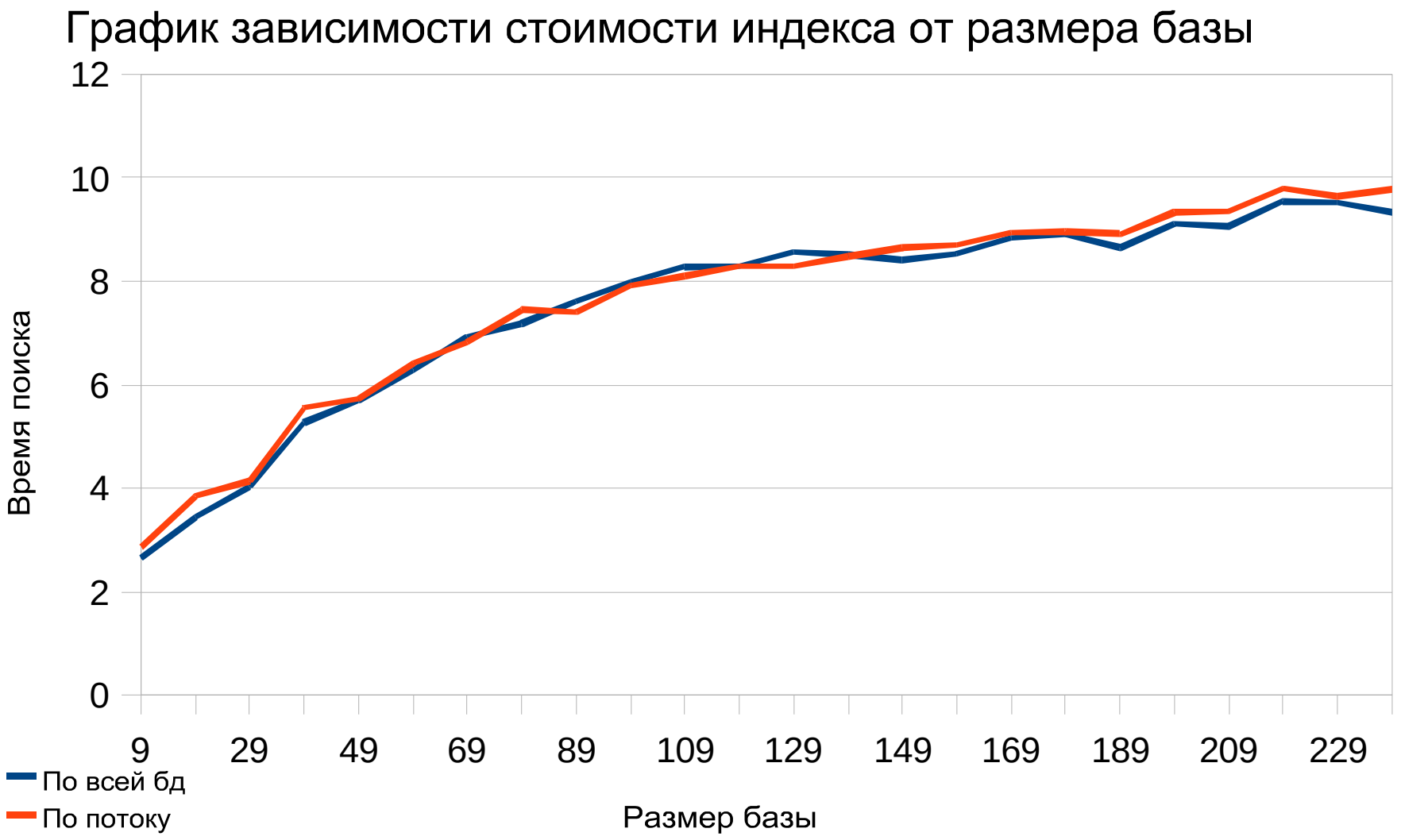
Оценки сложности отдельных операций для поиска в OEM-документах по CRP-запросам:

$$C_{|S|}= O(|S|) \quad C_{S1+S2} = O(|S|^2) \quad C_{P\{S\}} = O(|S|^2)$$

OEM. Зависимость времени индексации от размера базы.



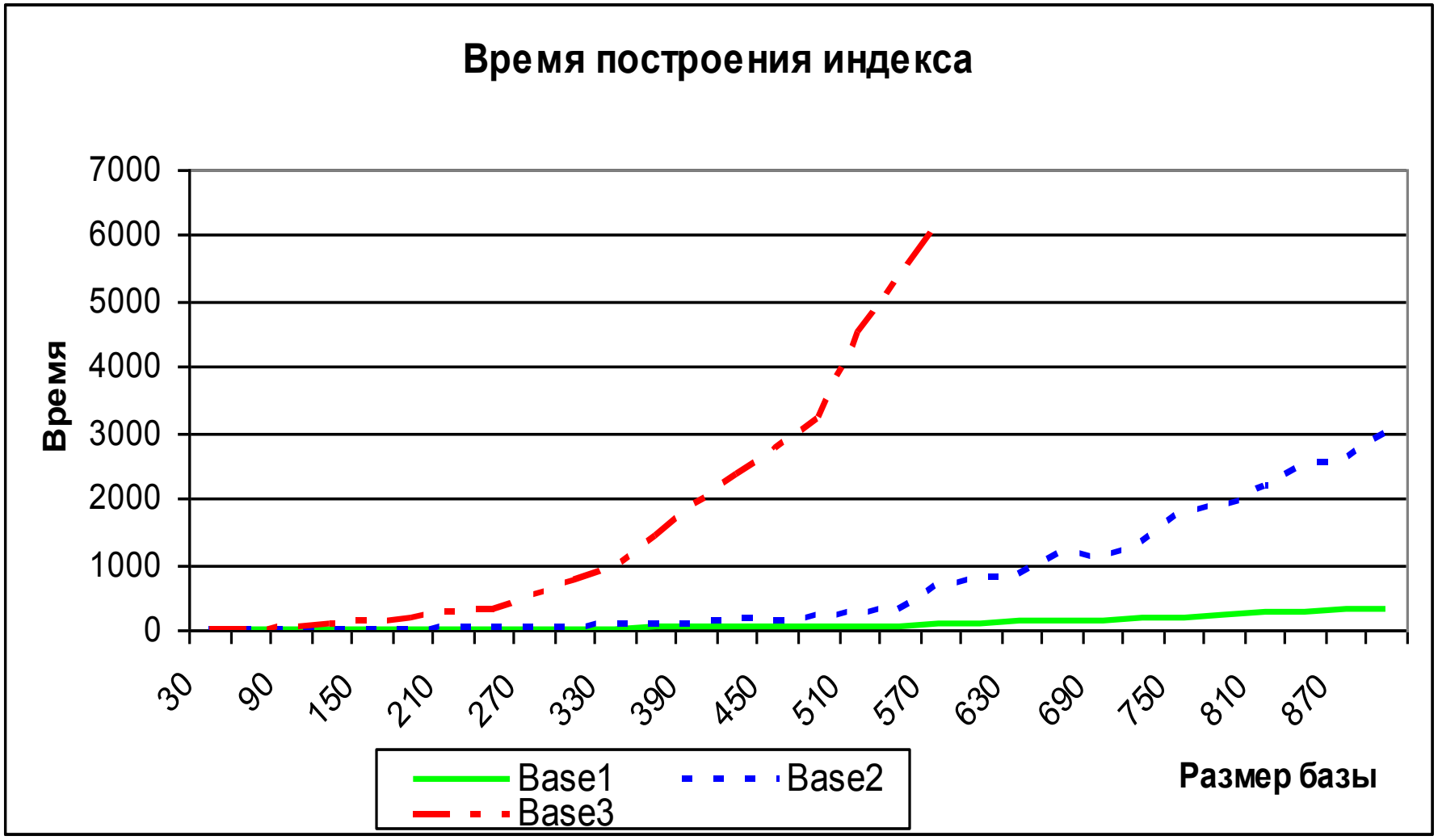
ОЕМ. Зависимость стоимости индекса от размера базы.



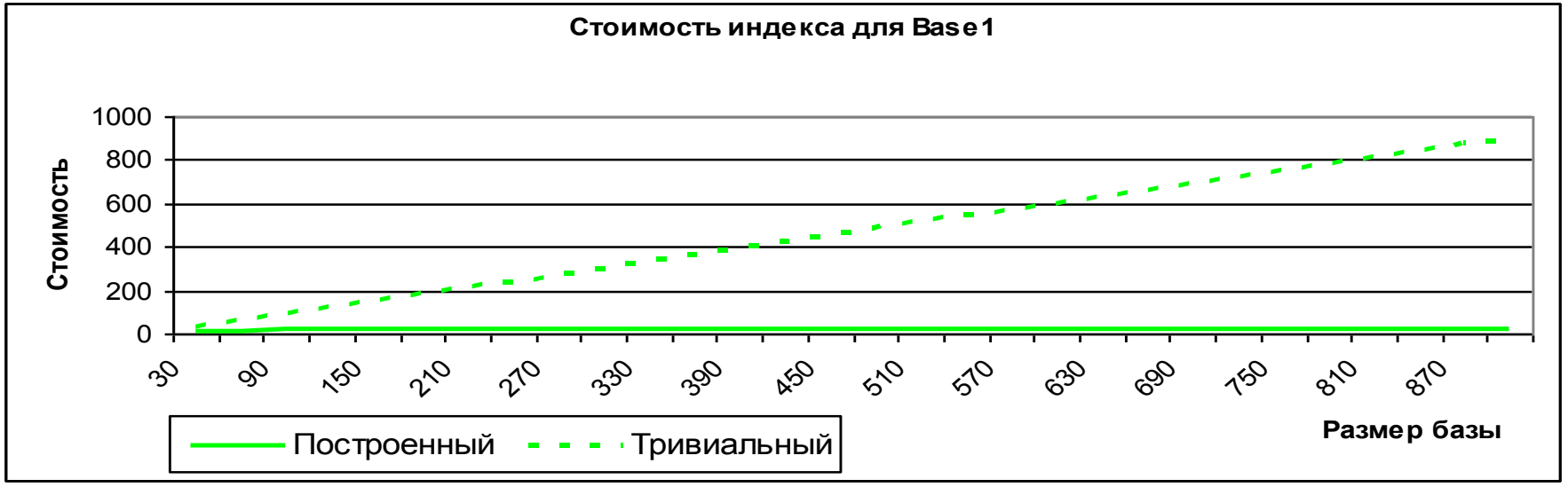
OEM. Поиск по запросам, состоящим из 1 символа в базе из 500 документов

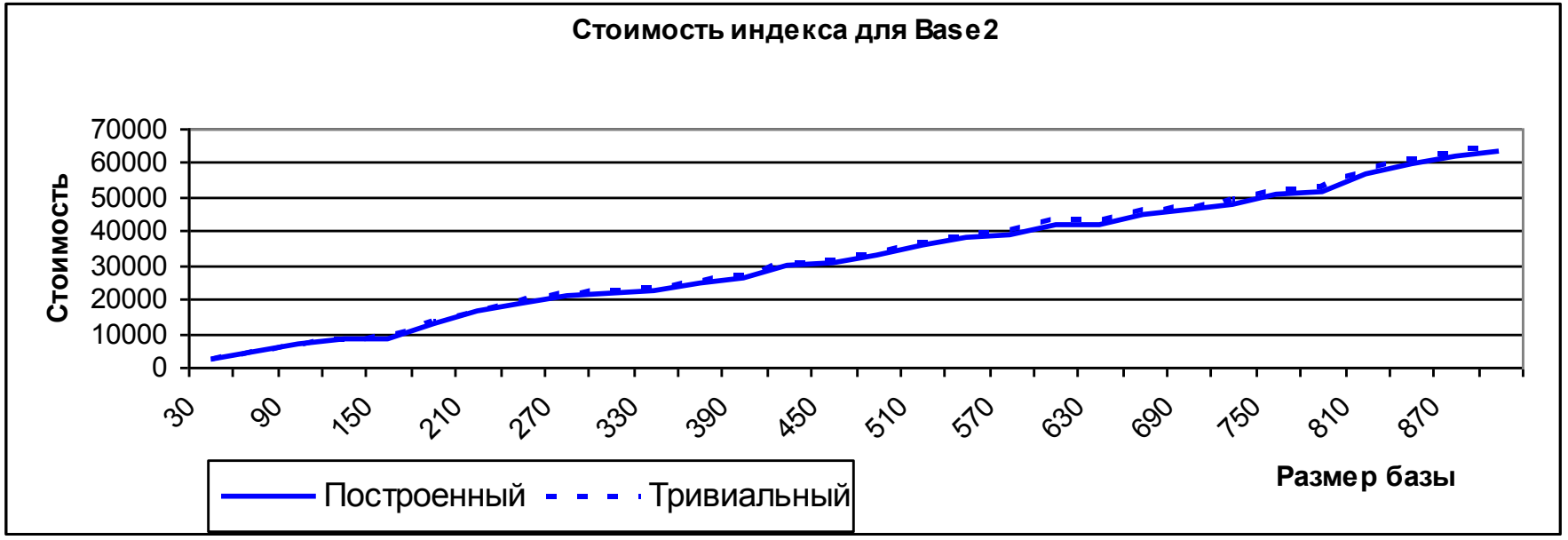
Количество запросов	4190
Количество документов	500
Суммарное время оптим. поиска	32,14
Суммарное время простого поиска	1771,16
Суммарное время мин. поиска	12,67
Среднее время оптим. поиска	0,008
Среднее время простого поиска	0,42
Среднее время мин. поиска	0,003
Коэффициент ускорения	55,102
Макс. возможный коэфф. ускорения	139,79

XML. Зависимость времени индексации от размера базы.

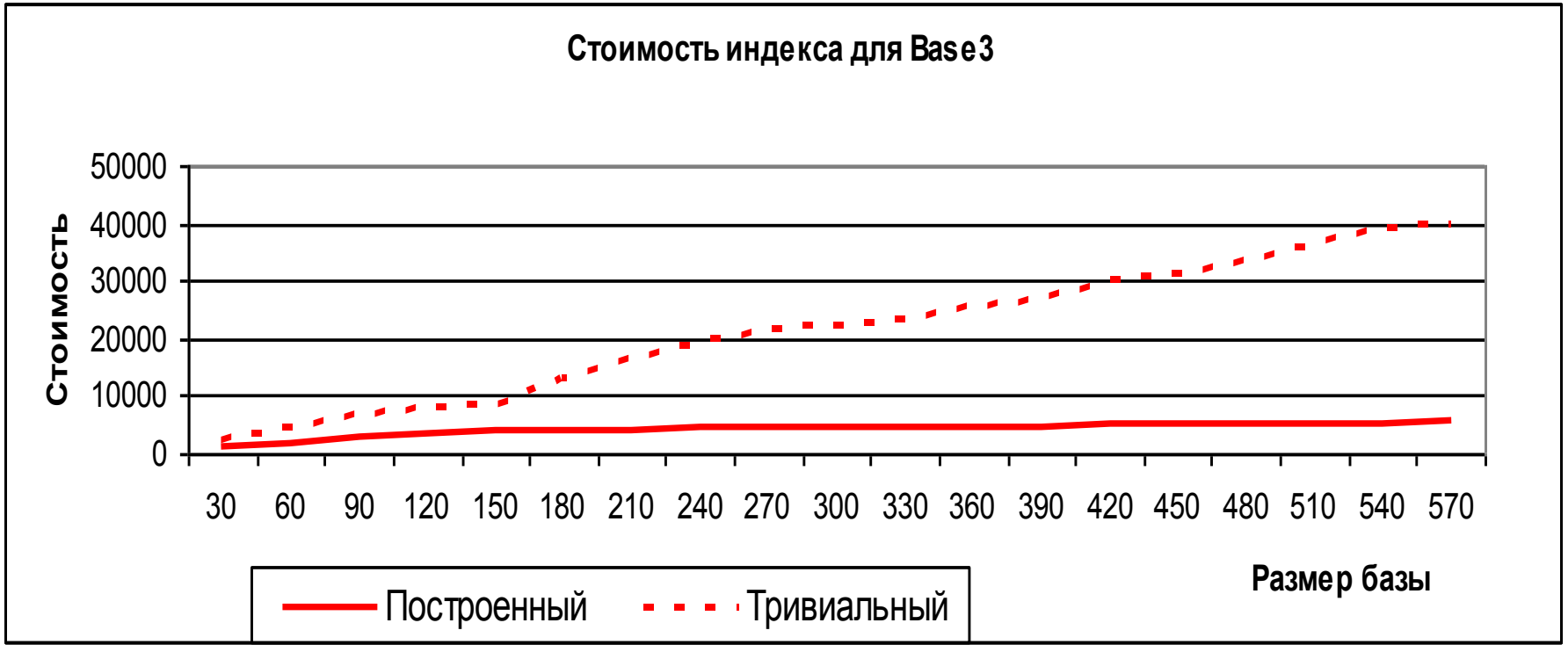


XML. Зависимость стоимости индекса от размера базы.





XML. Зависимость стоимости индекса от размера базы.



Оценка сложности поиска снизу

- Предположим, что мощность множества запросов равна M, все запросы равновероятны, при этом все результаты их вычисления различаются между собой.
- Тогда не существует такой иерархии схем, средняя вычислительная сложность поиска по которой меньше $\ln_2(M)$. При этом, существует алгоритм поиска, сложность которого $O(\ln_2(M))$.

Пример неэффективности иерархических индексов.

- D = натуральные числа.
- S = конечные интервалы натуральных чисел.
- Q = конечные множества натуральных чисел.

- Количество результатов поиска = 2^N.
- Нижняя оценка сложности поиска использованием индекса = $\ln_2(2^N) = N$.
- Следовательно, использование индекса в такой модели не может существенно снизить сложность поиска.

Спасибо за внимание