Функциональная сложность задачи подсчёта для двумерной задачи о доминировании

А. П. Пивоваров

В работе рассматривается задача подсчёта для двумерной задачи о доминировании. В рамках информационно-графовой модели данных приводится три семейства алгоритмов, решающих данную задачу. Оцениваются объёмы требуемой памяти и время обработки запроса в худшем случае. В качестве одного из средств уменьшения времени обработки запросов за счёт увеличения объёма требуемой памяти используется техника частичного каскадирования.

Ключевые слова: двумерная задача о доминировании, вычислительный информационный граф, информационный граф, функциональная сложность, частичное каскадирование.

1. Введение

В данной работе рассматривается задача подсчёта для двумерной задачи о доминировании. Её можно вкратце описать следующим образом. Задана некоторая коммутативная полугруппа $\mathcal{Z}$ и база данных, состоящая из точек, лежащих на единичном квадрате, причём каждой точке приписан некоторый вес — элемент полугруппы $\mathcal{Z}$. Запрос имеет вид точки на квадрате, а ответ на запрос равен результату применения полугрупповой операции между весами всех элементов базы, которые не превосходят запрос по обеим координатам. Рассматриваемая задача является модификацией обыкновенной (перечислительной) двумерной задачи о доминировании, где требуется перечислить сами записи, не превосходящие запрос по каждой из координат.

Подобные задачи, включая задачу двумерного интервального поиска, рассматривались у многих авторов. В [1] приводится алгоритм дерева регионов для решения двумерной задачи интервального поиска. Также в [1] приводится модификация Уиларда-Люкера для этого алгоритма, идея которой схожа с главной идеей техники частичного каскадирования, изложенной Чазелле и Гуибасом в [2]. В [3] для всё той же задачи двумерного интервального поиска строится семейство алгоритмов её решения по сути соответствующее первому из семейств, исследуемых в данной работе.

При этом в подавляющем большинстве работ рассматриваются порядковые либо асимптотические оценки характеристик алгоритма. Этот факт затрудняет использование полученных алгоритмов, поскольку то, что один алгоритм имеет лучшие характеристики в пределе (при размере базы данных стремящейся к бесконечности) ещё не значит, что он лучше при конкретном размере базы.

В данной же работе для представленных алгоритмов вычисляются конкретные оценки на объём требуемой памяти и время работы в худшем случае. Это позволяет к тому же оценить сверху сложность решения задач с базами данных определённого размера в зависимости от доступного объёма памяти. В качестве модели для рассмотрения алгоритмов решения задачи и обсчёта их характеристик используется модификация информационно-графовой модели данных для вычислительных задач информационного поиска (см. [4]).

Автор выражает благодарность и признательность научному руководителю профессору Э. Э. Гасанову.

2. Определение вычислительного информационного графа (ВИГ)

Определение вычислительного информационного графа (ВИГ) совпадает с определением ВИГ, предложенным в [4]. Для полноты изложения мы приведём определение понятия ВИГ в данной работе.

Тип вычислительных задач информационного поиска (ВЗИП) представляет собой пятерку $\langle X, Y, Z, \rho, \xi \rangle$, где X — множество запросов, Y — множество всех возможных записей, Z представляет из себя множество ответов, ρ — отношение поиска, заданное на $X \times Y$. $\xi : 2^Y \to Z$ — функция ответа, определённая на 2^Y — множестве подмножеств Y (достаточно, чтобы ξ была определена для всех конечных подмножеств Y). Сама задача информационного поиска представляет из себя пятерку $I = \langle X, V, Z, \rho, \xi \rangle$, где V — конечное подмножество Y. Содержательно задача поиска будет пониматься в том, чтобы для заданного запроса $x \in X$ находить такой ответ $z \in Z$, что $z = \xi\left(\{y \in V : x \,\rho\, y\}\right)$. Будем обозначать этот ответ как $z_I(x) = \xi\left(\{y \in V : x \,\rho\, y\}\right)$ и называть *правильным ответом для задачи I на запрос x.*

Для определения понятия ВИГ нам понадобятся следующие множества:

- множество запросов X;
- *множество F одноместных предикатов*, заданных на множестве X;
- *множество G одноместных переключателей*, заданных на множестве X (*переключатели* — это функции, область значений которых является начальным отрезком натурального ряда);
- множество ответов Z;
- множество состояний ВИГ M с выделенным элементом $m_0 \in M$, называемым *начальным состоянием*, кроме того зафиксируем *функцию вычисления ответа по состоянию* $\sigma : M \to Z$;
- *множество H функций изменения состояния вида $h : M \to M$, такое, что любые две функции из H коммутируют относительно операции суперпозиции.*

Шестерку $\mathcal{F} = \langle F, G, H, M, m_0, \sigma \rangle$ будем называть *базовым множеством.*

Определение понятия ВИГ разбивается на два шага. На первом шаге раскрывается структурная часть этого понятия, на втором — функциональная.

Определение ВИГ с точки зрения его структуры.

Пусть нам дана ориентированная многополюсная сеть.

Выделим в ней один полюс и назовем его *корнем*, а остальные полюса назовем *листьями*.

Выделим в сети некоторые вершины и назовем их *точками переключения* (полюса могут быть точками переключения).

Если β — вершина сети, то через ψ_β обозначим *полустепень исхода* вершины β.

Каждой точке переключения β сопоставим некий символ из G. Это соответствие назовем *нагрузкой точек переключения*.

Для каждой точки переключения β ребрам, из нее исходящим, поставим во взаимно однозначное соответствие числа из множества $\{1, 2, \ldots, \psi_\beta\}$. Эти ребра назовем *переключательными*, а это соответствие — *нагрузкой переключательных ребер*.

Ребра, не являющиеся переключательными, назовем *предикатными*.

Каждому предикатному ребру сети сопоставим некоторый символ из множества F. Это соответствие назовем *нагрузкой предикатных ребер*.

Сопоставим каждому листу сети некоторый элемент из множества H. Это соответствие назовем *нагрузкой листьев*.

Полученную нагруженную сеть назовем *вычислительным информационным графом* над базовым множеством $\mathcal{F} = \langle F, G, H, M, m_0, \sigma \rangle$.

Определение функционирования ВИГ.

Скажем, что предикатное ребро проводит запрос $x \in X$, если предикат, приписанный этому ребру, принимает значение 1 на запросе x; переключательное ребро, которому приписан номер n, проводит запрос $x \in X$, если переключатель, приписанный началу этого ребра, принимает значение n на запросе x; ориентированная цепочка ребер проводит запрос $x \in X$, если каждое ребро цепочки проводит запрос x; запрос $x \in X$ проходит в вершину β ВИГ, если существует ориентированная цепочка, ведущая из корня в вершину β, которая проводит запрос x.

Пусть $\alpha_1, \ldots, \alpha_p$ — это все листья, в которые проходит запрос x. Пусть этим листьям приписаны функции $h_1, \ldots, h_p$ соответственно. Назовем состояние $m_U(x) = h_1 \circ \cdots \circ h_p(m_0)$ *финальным состоянием* ВИГ U на запросе x. Если запрос x не проходит ни в один лист, положим $m_U(x) = m_0$. Будем называть $z_U(x) = \sigma(m_U(x))$ *функцией ответа ВИГ* U.

Понятие ВИГ полностью определено.

Пусть β — некоторая вершина ВИГ. Тогда обозначим $\varphi_\beta(x)$ предикат на X, такой что $\varphi_\beta(x) = 1$ для тех и только тех запросов x, которые проходят в вершину β. Предикат $\varphi_\beta(x)$ называется *функцией фильтра вершины* β.

Будем говорить, что ВИГ U решает ВЗИП $I = \langle X, V, Z, \rho, \xi \rangle$ если для любого запроса $x \in X$ имеем $z_U(x) = z_I(x)$, где $z_I(x) = \xi\left(\{y \in V : x\,\rho\,y\}\right)$.

Каждому ВИГ U поставим в соответствие следующую процедуру поиска. Процедура получает на вход запрос x и выдаёт элемент $z \in Z$, называемый *ответом процедуры на запрос* x. Процедура может оставлять пометки на вершинах ВИГ. Кроме того, используется промежуточное множество вершин A. Также имеется состояние ВИГ $m \in M$. В начале m устанавливается равным m_0.

Начнём описание самой процедуры. Корень ВИГ U помечается и добавляется в множество A. Процедура рассматривает элементы A по одному до тех пор пока A не пусто. Для каждой рассматриваемой вершины β множества A, процедура делает следующее:

- Если вершина β является листом, то рассматривается функция изменения состояния $h \in H$, приписанная листу β. Эта функция применяется к текущему состоянию m, то есть мы устанавливаем $h(m)$ в качестве текущего состояния ВИГ;
- Если β — переключательная вершина, то процедура вычисляет значение переключателя $g(x)$, приписанного вершине β. Пусть $g(x) = n$, число n приписано некоторому переключательному ребру (β, γ) и вершина γ не помечена. Тогда вершина γ помечается и добавляется к A;
- Если β — не переключательная вершина, то рассматриваются все ребра, исходящие из β. Для каждого ребра (β, γ), исходящего из β процедура вычисляет предикат $f(x)$, приписанный предикатному ребру (β, γ). Если $f(x) = 1$ и вершина γ не помечена, то вершина γ помечается и добавляется к A;
- Вершина β исключается из множества A.

ВИГ U к моменту завершения описанных действий будет находиться в состоянии m, которое, как не трудно заметить, равно $m_U(x)$. После этого процедура вычисляет $\sigma(m)$. Это значение процедура и выдаёт. Очевидно, ответ процедуры на запрос x равен $z_U(x)$ для любого запроса x.

Сложностью ВИГ U на запросе x назовем число $T(U,x)$, равное сумме числа переключателей и предикатов, вычисленных в процессе обработки запроса x. Эту величину также называют временем работы U на запросе x. *Сложностью* (или временем работы в худшем случае) называют величину $T(U) = \max_{x\in X} T(U,x)$.

Объемом $Q(U)$ ВИГ U назовём число рёбер в U. Эта величина характеризует объём памяти, требуемый для реализации алгоритма поиска, задаваемого ВИГ U.

3. Формулировка задачи и основные результаты

Пусть $\mathcal{Z} = (Z, \bigoplus)$ — коммутативная полугруппа с нулём. Операцию $\bigoplus$ будем для простоты будем именовать *суммой*. Рассмотрим тип вычислительных задач информационного поиска $S^{\mathcal{Z}}_{dom2} = \left\langle X_{dom2}, Y^{\mathcal{Z}}_{dom2}, Z, \rho_{dom2}, \xi^{\mathcal{Z}}_{dom2}\right\rangle$, где $X_{dom2} = [0,1]^2$, $Y^{\mathcal{Z}}_{dom2} = [0,1]^2 \times Z$, отношение поиска ρ_{dom2} определено таким образом, что для любых $x = (x_1, x_2) \in X_{dom2}$ и $y = (y_1, y_2, y_*) \in Y^{\mathcal{Z}}_{dom2}$ соотношение $x\, \rho_{dom2}\, y$ справедливо тогда и только тогда, когда одновременно выполнены условия $x_1 \geqslant y_1$ и $x_2 \geqslant y_2$. Элементы множества записей Y можно рассматривать как точки на плоскости, которым дополнительно приписаны веса из Z. Функция ответа $\xi^{\mathcal{Z}}_{dom2} : 2^Y \to Z$ определена на конечных подмножествах множества Y следующим образом (значение на бесконечных подмножествах Y определять не требуется): пусть $V = \{y^1, \dots, y^k\} \subset Y$ и $y^i = (y^i_1, y^i_2, y^i_{\mathcal{Z}})$, тогда $\xi^{\mathcal{Z}}_{dom2}(V) = y^1_{\mathcal{Z}} \bigoplus \dots \bigoplus y^k_{\mathcal{Z}}$. Отдельно определим $\xi^{\mathcal{Z}}_{dom2}(\varnothing)$ равным нейтральному элементу $\mathcal{Z}$.

Вычислительный информационный граф (ВИГ) для решения задач типа $S^{\mathcal{Z}}_{dom2}$ будем строить над базовым множеством

$\mathcal{F}^{\mathcal{Z}}_{dom2} = \langle F_{dom2}, G_{dom2}, H_{\mathcal{Z}}, M_{\mathcal{Z}}, m^{\mathcal{Z}}_0, \sigma_{id}\rangle$. Здесь $F_{dom2} = \big\{ f^i_{\geqslant a}\colon a \in [0,1],$ $i \in \{1,2\}\big\}$, где для любого $x = (x_1, x_2) \in X$ значение $f^i_{\geqslant a}(x)$ равно 1 тогда и только тогда, когда $x_i \geqslant a$. Предикат $f^1_{\geqslant 0}$, значение которого равно 1 на любом запросе x, будем называть *тождественно единичным предикатом.* Множество переключателей $G_{dom2} = \big\{g^i_{<a}\colon a \in [0,1], i \in \{1,2\}\big\}$, где для любого запроса $x = (x_1, x_2) \in X$ имеет место $g^i_{<a}(x) = \begin{cases} 1, & \text{если } x_i < a; \\ 2, & \text{если } x_i \geqslant a. \end{cases}$ Множество состояний ВИГ $M_{\mathcal{Z}}$ совпадает со множеством элементов рассматриваемой полугруппы (группы) Z, и начальное состояние $m^{\mathcal{Z}}_0$ — нейтральный элемент группы $\mathcal{Z}$, множество функций, меняющих состояние $H_{\mathcal{Z}} = \{h_z\colon h_z(b) = b \bigoplus z \ \forall b \in Z\}_{z\in Z}$ и функция выдачи ответа $\sigma_{id}(z) = z$ для любых состояний $z \in Z$.

Пусть U — некоторый ВИГ над базовым множеством $\mathcal{F}$. Пусть $I = \big\langle X_{dom2}, V, Z, \rho_{dom2}, \xi^{\mathcal{Z}}_{dom2}\big\rangle \in S^{\mathcal{Z}}_{dom2}$ — вычислительная задача информационного поиска ($V \subset Y^{\mathcal{Z}}_{dom2}$). Пусть $\mathcal{U}(I, \mathcal{F})$ — множество ВИГ над базовым множеством $\mathcal{F}$, решающих задачу I. Сложностью задачи I при базовом множестве $\mathcal{F}$ и объёме q назовём число

$$T(I, \mathcal{F}, q) = \min\{T(U)\colon U \in \mathcal{U}(I, \mathcal{F}) \text{ и } Q(U) \leqslant q\}.$$

Если k — натуральное число, $S = \langle X, Y, Z, \rho, \xi\rangle$ — тип вычислительных задач информационного поиска, то обозначим

$$\mathcal{I}(k, S) = \{I = \langle X, V, Z, \rho, \xi\rangle \in S\colon |V| = k\}.$$

Будем исследовать функцию, характеризующую зависимость сложности задач класса ВЗИП $\mathcal{I}(k, S^{\mathcal{Z}}_{dom2})$ от объёма памяти, доступной для алгоритма решения:

$$\mathcal{T}(k, S^{\mathcal{Z}}_{dom2}, \mathcal{F}^{\mathcal{Z}}_{dom2}, q) = \max_{I \in \mathcal{I}(k, S^{\mathcal{Z}}_{dom2})} T(I, \mathcal{F}^{\mathcal{Z}}_{dom2}, q).$$

Теорема 1. *Пусть $\mathcal{Z} = (Z, \bigoplus)$ — коммутативная полугруппа (либо группа) с нулём. Тогда*

1) *Для любых натуральных $k \geqslant 3$ и s таких, что $2 \leqslant s < k$, имеет место*

$$\mathcal{T}\left(k, S^{\mathcal{Z}}_{dom2}, \mathcal{F}^{\mathcal{Z}}_{dom2}, \frac{2k\log_2 k}{\log_2 s} + 10k + 6\frac{k-1}{s-1}\right) \leqslant$$
$$\leqslant \frac{s}{2\log_2 s}\log_2^2 k + 4s\frac{\log_2 k}{\log_2 s} + \log_2 k + (\log_2 5 - 1)\frac{\log_2 k}{\log_2 s} + (1+\log_2 3)s.$$

2) *Для любых натуральных $k \geqslant 3$, s таких, что $2 \leqslant s < k$ и нечётного g', такого что $g' \geqslant 5$ если $s = 2$ и $g' \geqslant 9$ если $s \geqslant 3$, имеет место*

$$\mathcal{T}\left(k, S^{\mathcal{Z}}_{dom2}, \mathcal{F}^{\mathcal{Z}}_{dom2}, \frac{5(1+c')k\log_2 k}{\log_2 s} + (9+20c')k + 6\frac{k-1}{s-1}\right) \leqslant$$
$$\leqslant \frac{\log_2 k}{\log_2 s}\left(\frac{1}{2}\log_2 k + \log_2 15 + (s-1)(1 + \lceil\log_2(g'+1)\rceil)\right) +$$
$$+ 2\log_2 k + 2s + 1,$$

где $c' = \frac{4}{g'-3}$ *если* $s = 2$ *и* $c' = \frac{8}{g'-7}$ *если* $s \geqslant 3$.

3) *Для любых натуральных $k \geqslant 4$ и s таких, что $3 \leqslant s < k$ и нечётного $g'' \geqslant 17$, имеет место*

$$\mathcal{T}\Big(k, S^{\mathcal{Z}}_{dom2}, \mathcal{F}^{\mathcal{Z}}_{dom2}, \frac{7(1+c'')k\log_2 k}{\log_2 s} + 14k -$$
$$- 8 + 8\frac{k-1}{s-1} + 28c''(7k - 7 + 4\frac{k-1}{s-1})\Big) \leqslant$$
$$\leqslant \frac{\log_2 k}{\log_2 s}(s + \lceil\log_2(2s+1)\rceil)(1 + \lceil\log_2(g''+1)\rceil) + 2s,$$

где $c'' = \frac{16}{g''-15}$.

4) *Для любых натуральных $k \geqslant 3$ и любого нечётного $g'' \geqslant 13$, имеет место*

$$\mathcal{T}\big(k, S^{\mathcal{Z}}_{dom2}, \mathcal{F}^{\mathcal{Z}}_{dom2}, 6(1+c'')k\log_2 k +$$
$$+ 22k - 16 + 264c''(k-1)\big) \leqslant$$
$$\leqslant 5\log_2 k\left(1 + \lceil\log_2(g''+1)\rceil\right) + 4,$$

где $c'' = \frac{12}{g''-11}$.

4. Понятие частичного каскадирования

При доказательстве основного результата нам потребуется воспользоваться техникой частичного каскадирования. Конкретно, нам будут нужны определения структур и оценки на их размеры из работы [5]. Приведём только формулировки используемых утверждений в несколько упрощённом виде и без доказательств.

Под *расширенными действительными числами* будем понимать элементы множества $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. *Каталогами* будем называть неубывающие последовательности расширенных действительных чисел. Один каталог будем называть *подкаталогом* другого, если он как последовательность является подпоследовательностью второго. Пусть C — каталог. Тогда каталог, получающийся из него удалением дубликатов будем обозначать $U(C)$.

Пусть есть два каталога $C = \{c_i\}_{i=1}^{n}$ и $D = \{d_j\}_{j=1}^{m}$. Пару индексов $p = (i, j)$, такую что $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$ будем называть *переходом* между каталогами C и D, если $c_i = d_j$. Про элементы c_i и d_j будем говорить, что они *принадлежат переходу* p. Будем называть некоторый переход x-*переходом*, если значение соединяемых этим переходом элементов равно x.

Последовательность переходов $B = \{p_k = (i_k, j_k)\}_{k=1}^{s}$ где $s \geqslant 2$ будем называть *мостом* если номера элементов и в первом и во втором каталоге строго возрастают.

Будем говорить, что элемент c_i каталога C *принадлежит k-му ярусу моста* B $(1 \leqslant k \leqslant s-1)$ в том случае, если $i_k < i < i_{k+1}$. Каждый элемент каталога C принадлежит не более чем одному ярусу моста B, но может не принадлежать ни одному ярусу если он принадлежит одному из переходов моста, либо лежит выше верхнего перехода моста, либо лежит ниже нижнего перехода моста. Аналогично, будем говорить, что элемент d_j каталога D принадлежит k-му ярусу моста B, если $j_k < j < j_{k+1}$.

Шириной k-го яруса моста B будем называть суммарное количество элементов в каталогах C и D, принадлежащих k-му ярусу моста B. *Шириной моста* B будем называть максимальную ширину некоторого яруса данного моста.

Отрезок $[c_{i_1}, c_{i_s}]$ $(= [d_{j_1}, d_{j_s}])$ будем называть *отрезком покрытия моста* B.

Графом каталогов будем называть пару $L = (G, \mathcal{C})$, где $G = (V, E)$ — граф без петель и кратных ребер (V — множество вершин графа G, а E множество рёбер G), $\mathcal{C} = \{C_v\}_{v \in V}$ — набор каталогов, поставленных в соответствие вершинам G.

Степенью графа каталогов $L = (G, \mathcal{C})$ будем называть величину $d(L)$, равную максимальной из всех степеней вершин графа G.

Системой мостов будем называть тройку $S = (G, \mathcal{A}, \mathcal{B})$, где $(G, \mathcal{A})$ образуют граф каталогов ($\mathcal{A} = \{A_v\}_{v \in V}$ — некоторый набор каталогов), а $\mathcal{B} = \{B_e\}_{e \in E}$ — набор мостов, такой что выполнены условия:

1) для любого ребра $\{v, w\} = e \in E$ мост B_e является мостом между каталогами A_v и A_w, либо между каталогами A_w и A_v;
2) для любого ребра $e \in E$ отрезок покрытия моста B_e равен $[-\infty, \infty]$;
3) для любого $x \in \overline{\mathbb{R}}$ каждый мост из $\mathcal{B}$ содержит не более одного x-перехода.

Шириной системы мостов $S = (G, \mathcal{A}, \mathcal{B})$ будем называть максимальную ширину моста среди $\mathcal{B}$.

Система мостов $S = (G, \mathcal{A}, \mathcal{B})$ называется *согласованной* с графом каталогов $L = (G, \mathcal{C})$ если для любого $v \in V$ каталог $U(C_v)$ является подкаталогом A_v.

Лемма 1 (о частичном каскадировании). *Пусть имеется граф каталогов $L = (G, \mathcal{C})$, где $G = (V, E)$ и $|V| = n$, $|E| = m$. Пусть $d = d(L)$ — степень графа каталогов L. Пусть зафиксирован нечётный натуральный параметр $g \geqslant 4d + 1$.*

Тогда существует система мостов $S = (G, \mathcal{A}, \mathcal{B})$, согласованная с графом каталогов L и при этом выполнены следующие условия:

(а) ширина системы мостов S не превосходит параметра g;
(б) имеет место неравенство $\sum_{v \in V} |A_v| \leqslant (1 + c) \sum_{v \in V} |U(C_v)| + 2n + 4mc \leqslant (1 + c) \sum_{v \in V} |C_v| + 2n + 4mc$, где $c = \frac{4d}{g+1-4d}$;

(в) *для любого $v \in V$ значение каждого из элементов A_v либо равно значению некоторого элемента из C_w для некоторого $w \in V$, либо равно $\pm\infty$;*

(г) *любой каталог A_v содержит не более одной записи с некоторым значением $x \in \overline{\mathbb{R}}$.*

Приведённая выше лемма о частичном каскадировании является частным случаем одного из результатов, полученных в работе [5].

5. Доказательство основного результата

Будем строить вычислительные информационные графы, решающие некоторую задачу $I = \left\langle X_{dom2}, V, Z, \rho_{dom2}, \xi^{\mathcal{Z}}_{dom2} \right\rangle \in S^{\mathcal{Z}}_{dom2}$. Будем обозначать $|V|$ буквой k. Рассматривается только случай, когда $k \geqslant 3$.

Определим одно простое, но очень важное построение, которое нам в дальнейшем потребуется. Пусть в процессе построения ВИГ у нас была объявлена некоторая вершина v и последовательность вершин $v_0, v_1, \ldots, v_n$ $(n \geqslant 1)$, и есть последовательность действительных чисел $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$ (вершине v_0 не соответствует никакое из чисел a_i). Определим операцию *построения дихотомического дерева по i координате* $(i \in \{1, 2\})$, *основанного на вершинах $v_0, v_1, \ldots, v_n$ со значениями $a_1, \ldots, a_n$ и корнем в вершине v* следующим образом. Считаем вершины $v_0, v_1, \ldots, v_n$ расположенными подряд слева направо и выпускаем из v бинарное дерево минимально возможной высоты, концевыми вершинами которого являются в точности вершины $v_0, v_1, \ldots, v_n$ и такое, что каждая его внутренняя вершина имеет ровно двух потомков. Все внутренние вершины построенного дерева (включая корневую вершину v) объявляются переключательными и каждой внутренней вершине w приписывается переключатель $g^i_{<a_{j(w)}}$, где $j(w) = \min\{j \colon v_j$ лежит в правом поддереве $w\}$. Очевидно, что для любой внутренней вершины w значение $1 \leqslant j(w) \leqslant n$. Левому ребру, исходящему из w приписывается номер 1, правому — 2. Для построенной части ВИГ выполняются следующие условия:

- количество добавленных рёбер равно $2n$;

- если некоторый запрос $x = (x_1, x_2)$ попадает в вершину v, то внутри дерева он проходит по одной из его ветвей до некоторой одной вершины v_j, такой что $a_j \leqslant x_i < a_{j+1}$, где для удобства считаем $a_0 = -\infty$, $a_{n+1} = \infty$. При этом количество вычисленных при проходе по ветви переключателей не превосходит $\lceil \log_2(n+1) \rceil$. Здесь и далее под $\lceil t \rceil$ будем понимать число t, округленное вверх до ближайшего целого.

Для построения ВИГ, решающего поставленную задачу, нам потребуются некоторые стандартные графы, которые будут использоваться как составные части. Пусть V' — некоторое непустое подмножество V. Тогда графом $U^2_{V'}$ будем называть граф, который для запроса $x = (x_1, x_2) \in [0,1]^2$ находит сумму весов всех таких записей $y = (y_1, y_2, y_*) \in V'$, что $y_2 \leqslant x_2$. Пусть $V' = \{y^1, \dots, y^t\}$, где $t = |V'|$ и запись $y^j = (y^j_1, y^j_2, y^j_*)$, причём записи упорядочены по неубыванию второй координаты. Граф $U^2_{V'}$ строится так. Объявляется корневая вершина v и концевые вершины $v_0, v_1, \dots, v_t$. Производится построение дихотомического дерева по 2 координате, основанного на вершинах $v_0, v_1, \dots, v_t$ со значениями $y^1_2, \dots, y^t_2$ и корнем в вершине v. Для каждой концевой вершины v_j с ненулевым номером j вычисляется сумма весов $z_j = y^1_* \bigoplus \cdots \bigoplus y^j_*$ и вершине v_j приписывается функция изменения состояния $h_{z_j} \in H_{\mathcal{Z}}$. Несложно видеть, что построенное дерево $U^2_{V'}$ решает поставленную задачу и имеет характеристики $Q(U^2_{V'}) = 2t = 2|V'|$, $T(U^2_{V'}) \leqslant \lceil \log_2(|V'|+1) \rceil$.

Кроме $U^2_{V'}$ нам потребуется ввести граф $U^e_{V'}$ — переборный граф решения исходной задачи для множества $V' \subseteq V$. Этот граф строится совсем просто. Берётся корень и объявляется предикатной вершиной. Затем для каждой записи $y = (y_1, y_2, y_*) \in V'$ из корня выводим сначала предикатное ребро с предикатом $f^1_{\geqslant y_1}$ и затем из конца этого ребра другое предикатное ребро, которому приписан $f^2_{\geqslant y_2}$. Концу второго ребра приписываем функцию изменения состояния h_{y_*}. Полученный граф имеет характеристики $Q(U^e_{V'}) = 2|V'|$, $T(U^e_{V'}) \leqslant 2|V'|$.

Будем решать задачу I параметризованным методом, обобщающим метод дерева регионов. Зафиксируем некоторый целочисленный параметр s таким, чтобы $2 \leqslant s < k$ и построим дерево T^s_V по библиотеке V и параметру s следующим образом. Сначала упорядочим записи V в порядке возрастания первой координаты (порядок записей с одинаковыми первыми координатами не важен). Вершины дерева T_V^s будем индексировать с помощью слов в алфавите $\{1,\ldots,s\}$, то есть каждая вершина будет называться w_α, где $\alpha \in \{1,\ldots,s\}^*$. Корень дерева назовем w_λ, где λ — пустое слово. Каждой вершине w_α будем приписывать некоторое множество записей $V_\alpha \subseteq V$. Корню дерева T_V^s припишем множество $V_\lambda = V$. Затем, до тех пор пока в дереве есть концевая вершина w_α с приписанным ей множеством V_α, такая что $|V_\alpha| > s$, будем делать следующую процедуру: разобьём множество V_α на s непересекающихся подмножеств $V_{\alpha 1},\ldots,V_{\alpha s}$ одинакового размера (если $|V_\alpha|$ не делится на s, то несколько первых множеств будут содержать на один элемент больше, чем оставшиеся). Разбиение проведем таким образом, чтобы любой элемент множества с меньшим номером был меньше, чем любой элемент множества с большим номером (сравнение элементов относительно введённого выше отношения линейного порядка на V). Из вершины w_α выпустим s ребер по направлению к s новым вершинам, имена которых $w_{\alpha 1},\ldots,w_{\alpha s}$. Сопоставим новым вершинам множества $V_{\alpha 1},\ldots,V_{\alpha s}$ соответственно. В момент, когда каждой концевой вершине будет сопоставлено некоторое множество записей размера меньше или равного s, процесс остановится и мы получим дерево T_V^s. Количество $|V|$ записей в базе будем обозначать буквой k. Легко видеть, что количество уровней в построенном дереве равно $L = \lceil \log_s k \rceil$. Так как $s < k$, то $L \geqslant 2$. Заметим также, что построение T_V^s начинается с дерева, у которого одна концевая вершина и ни одной внутренней, а каждая операция достраивания делает одну концевую вершину внутренней и добавляет s новых концевых вершин. Поэтому у построенного дерева выполнено соотношение $t_{leaf} = (s-1)t_{in}+1$, где t_{leaf} — количество концевых вершин, а t_{in} — количество внутренних вершин. Но каждой концевой вершине T_V^s соответствует непустое подмножество записей V не пересекающееся с подмножествами, соответствующими другим концевым вершинам. Отсюда имеем:

$$t_{leaf} \leqslant k \quad \text{и} \quad t_{in} = \frac{t_{leaf}-1}{s-1} \leqslant \frac{k-1}{s-1}. \tag{1}$$

Дерево T_V^s будет в некотором смысле каркасом, на котором строятся все приведённые далее алгоритмы поиска. Идея алгоритмов будет везде следующей: для вычисления ответа по подбазе V_α эта самая база разбивается на s примерно одинаковых частей $V_{\alpha 1}, \ldots, V_{\alpha s}$, упорядоченных по первой координате. Для вычисления ответа на запрос необходимо проверить, какие из множеств $V_{\alpha 1}, \ldots, V_{\alpha s}$ по первой координате целиком лежат ниже запроса, какие целиком лежат выше и какие части лежат частично ниже, а частично выше запроса (всё по первой координате). Для частей $V_{\alpha i}$, целиком лежащих ниже запроса нужно взять сумму весов записей, лежащих ниже запроса по второй координате. Части $V_{\alpha i}$, целиком лежащие выше запроса можно дальше не рассматривать — ни одна запись из такой части не удовлетворяет запросу. Часть $V_{\alpha i}$, которая лежит частично ниже, а частично выше запроса может быть только одна (так как все части упорядочены по первой координате). Если размер такой части не превышает s, то ответ для неё строится с помощью перебора всех записей. Если же размер больше s, то приведённый выше алгоритм используется уже для подбазы $V_{\alpha i}$.

Рассмотрим вершину w_α дерева T_V^s, находящуюся на уровне с номером l, где под уровнем подразумевается расстояние от вершины до корня дерева. Легко убедиться, что размер соответствующего ей множества V_α можно оценить следующим образом: $|V_\alpha| \leqslant \lceil \frac{k}{s^l} \rceil$.

5.1. Обобщённый метод дерева регионов

Построим с помощью дерева T_V^s вычислительный информационный граф $U_I(s)$, решающий вычислительную задачу I. Для каждой вершины w_α дерева T_V^s заведём соответствующую ей вершину v_α строящегося информационного графа. Если w_α — концевая вершина T_V^s, то из v_α выпускаем граф $U_{V_\alpha}^e$, перебором решающий нашу задачу для множества записей V_α. Если же вершина w_α является внутренней (этот случай изображён на рисунке 1), то у неё есть s дочерних вершин $w_{\alpha 1}, \ldots, w_{\alpha s}$. Этим дочерним вершинам приписаны множества записей $V_{\alpha 1}, \ldots, V_{\alpha s}$ соответственно. Рассмотрим самое маленькое и самое большое значения первых координат записей

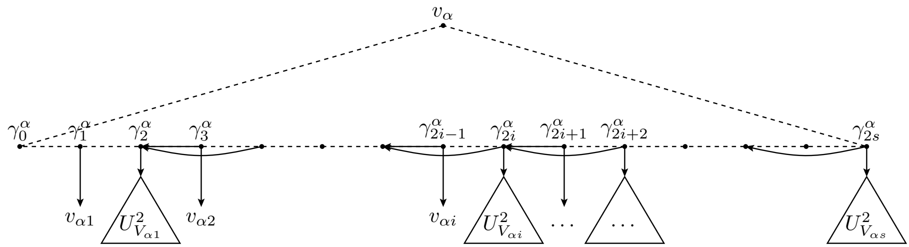

Рис. 1. Часть ВИГ $U_I(s)$, выпускаемая из вершины v_α, соответствующей внутренней вершине w_α дерева T_V^s. Сплошные рёбра на рисунке являются предикатными рёбрами с тождественно единичными предикатами.

из $V_{\alpha i}$ и обозначим эти значения c_{2i-1} и c_{2i} соответственно. По построению T_V^s выполняется $c_1 \leqslant c_2 \leqslant c_3 \cdots \leqslant c_{2s}$.

Объявим $2s+1$ новую вершину $\gamma_0^\alpha, \gamma_1^\alpha, \gamma_2^\alpha, \ldots, \gamma_{2s}^\alpha$ и проведём процедуру построения дихотомического дерева по 1 координате, основанного на вершинах $\gamma_0^\alpha, \gamma_1^\alpha, \gamma_2^\alpha, \ldots, \gamma_{2s}^\alpha$ со значениями $c_1, \ldots, c_{2s}$ и корнем в вершине v_α. Если некоторый запрос $x = (x_1, x_2)$, попавший в v_α, прошёл по дереву до некоторой вершины γ_j^α, возможны следующие варианты:

1) Пусть $j = 0$. Тогда имеет место $x_1 < c_1$ и ни один элемент базы V_α не удовлетворяет запросу x.
2) Пусть $j = 2i$ $(1 \leqslant i \leqslant s)$. Тогда имеет место $c_{2i} \leqslant x_1 < c_{2i+1}$ (если $i = s$, считаем $c_{2s+1} = \infty$). Тогда все подбазы $V_{\alpha t}$ при $t \leqslant i$ целиком удовлетворяют запросу x по первой координате, а записи из подбаз $V_{\alpha t}$ при $t > i$ не могут удовлетворять запросу x.
3) Пусть $j = 2i - 1$ — нечётное $(1 \leqslant i \leqslant s)$. Тогда имеет место $c_{2i-1} \leqslant x_1 < c_{2i}$. В этом случае мы должны найти решение задачи на подбазе $V_{\alpha i}$ и просуммировать его с решениями для всех подбаз $V_{\alpha t}$ для всех $t < i$, учитывая тот факт, что все записи $V_{\alpha t}$ при $t < i$ удовлетворяют запросу x по первой координате, а записи из подбаз $V_{\alpha t}$ при $t > i$ не могут удовлетворять запросу x.

Поэтому из каждой концевой вершины с чётным номером γ^α_{2i} выпустим предикатное ребро с тождественно единичным предикатом, ведущее в граф $U^2_{V_{\alpha i}}$. Если при этом $i > 1$, то выпустим из этой же вершины ещё одно предикатное ребро с тождественно единичным предикатом и концом в вершине с предшествующим чётным номером γ^α_{2i-2}. Таким образом, попадание в вершину γ^α_{2i} гарантирует прохождение запроса не только в подграф $U^2_{V_{\alpha i}}$, но и во все $U^2_{V_{\alpha t}}$ с $t < i$. Из каждой вершины с нечётным номером γ^α_{2i-1} выпустим предикат с тождественно единичным предикатом, ведущее в вершину $v_{\alpha i}$ — вершину, которая в строящемся ВИГ соответствует вершине $w_{\alpha i}$ дерева T^s_V. Плюс для каждой вершины с нечётным номером γ^α_{2i-1} при $i > 1$ выпускаем предикатное ребро с тождественно единичным предикатом в соседнюю вершину с чётным номером γ^α_{2i-2}. Таким образом, запрос, попавший в вершину v_α может либо попасть в вершину γ^α_{2i} и пройти по графам $U^2_{V_{\alpha t}}$ с $t \leqslant i$, либо попасть в вершину γ^α_{2i-1} и, кроме графов $U^2_{V_{\alpha t}}$ с $t < i$, пройти также в вершину $v_{\alpha i}$, лежащую на уровень ниже. Отметим здесь, что запрос может проходить только в одну из вершин с нечётным номером γ^α_{2i-1} и если это происходит, то запрос заведомо не может пройти в вершину γ^α_{2s}. На этом построение искомого ВИГ $U_I(s)$ завершается.

Подсчитаем объём ВИГ $U_I(s)$. Все переборные графы вида $U^e_{V'}$ строились по взаимно не пересекающимся множествам, объединение которых даёт всё V, поэтому количество ребер в них в сумме равно $2k$. Для каждой вершины $w_\alpha \in T^s_V$ кроме корня был построен один граф $U^2_{V_\alpha}$. Так как на одном уровне дерева T^s_V множества V_α не пересекаются, то это даёт нам дополнительно не более чем $(L-1)2k$ рёбер, где $L = \lceil \log_s k \rceil$ — количество уровней T^s_V. Для каждой внутренней вершины T^s_V дополнительно строилось дихотомическое дерево с $2s+1$ концевыми вершинами и, следовательно, количество ребер в нём равно $4s$. Из $2s-2$ концевых вершин дихотомического дерева выходило ещё по два ребра и из двух вершин выходило по одному ребру, что даёт нам ещё $4s-2$ ребра на внутреннюю вершину T^s_V. Итого по $8s-2$ ребра на внутреннюю вершину T^s_V. Так как количество внутренних вершин T^s_V не превосходит $\frac{k-1}{s-1}$, получаем:

$$Q(U_I(s)) \leqslant 2k+(L-1)2k+(8s-2)\frac{k-1}{s-1} < 2k\log_s k+10k+6\frac{k-1}{s-1}. \quad (2)$$

Подсчитаем теперь время работы $U_I(s)$ в худшем случае. Оценим сверху количество вычисленных операций: в худшем случае для каждого из $L-1$ уровня T_V^s от 0 до $L-2$ (уровень вершины равен расстоянию от неё до корня) производится прохождение по дихотомическому дереву с $2s+1$ концевыми вершинами, что даёт нам не более $\lceil \log_2(2s+1)\rceil \leqslant \lceil \log_2 \frac{5s}{2}\rceil < \log_2(5s)$ операций, затем запрос проходит не более чем в одну вершину с нечётным номером и не более чем в s вершин с чётными номерами (но если проходит в вершину с нечётным номером, то не проходит в вершину с самым большим чётным номером $2s$), где в сумме вычисляется не более чем $2s-1$ тождественных предикатов (так как посещается максимум $s-1$ вершина, из которой выпущено два ребра с тождественными предикатами и ещё одна вершина, из которой выпущено одно такое ребро). На каждом уровне производится не более s поисков в структурах вида $U^2_{V'}$, в каждой из которых $|V'| \leqslant \lceil \frac{k}{s^{l+1}}\rceil$, где l — номер рассматриваемого уровня, поэтому для каждой из этих структур количество операций не превосходит $\left\lceil \log_2\left(\lceil \frac{k}{s^{l+1}}\rceil + 1\right)\right\rceil$. Так как $L = \lceil \log_s k\rceil$, то $L < \log_s k + 1$ и $s^{L-1} < k$. Отсюда получаем, что при $l = L-2$ имеет место $\frac{k}{s^{l+1}} > 1$, а при $l < L-2$ имеет место $\frac{k}{s^{l+1}} > 2$. Получаем, что при $l = L-2$ количество добавляемых операций не больше $\left\lceil \log_2\left(\lceil \frac{k}{s^{L-1}}\rceil + 1\right)\right\rceil \leqslant \left\lceil \log_2\left(\frac{k}{s^{L-1}} + 2\right)\right\rceil \leqslant \left\lceil \log_2 3\frac{k}{s^{L-1}}\right\rceil \leqslant \log_2 \frac{k}{s^{L-1}}+1+\log_2 3$. Для $l < L-2$ количество добавляемых операций не превосходит $\left\lceil \log_2\left(\lceil \frac{k}{s^{l+1}}\rceil + 1\right)\right\rceil \leqslant \left\lceil \log_2\left(\frac{k}{s^{l+1}} + 2\right)\right\rceil \leqslant \left\lceil \log_2 2\frac{k}{s^{l+1}}\right\rceil \leqslant \log_2 \frac{k}{s^{l+1}} + 2$. В сумме по всем l от 0 до $L-2$ получим следующее ограничение на количество операций:

$$\left(\sum_{l=0}^{L-2} \log_2 \frac{k}{s^{l+1}} + 1 + \log_2 3 + 2(L-2)\right) s =$$

$$= \left((L-1)\log_2 k - \sum_{l=0}^{L-2}(l+1)\log_2 s + 2L - 3 + \log_2 3\right) s =$$

$$= \left((L-1)\log_2 k - \frac{L(L-1)}{2}\log_2 s + 2L - 3 + \log_2 3 \right) s <$$
$$< \left((L-1)(\log_2 k - \frac{\log_s k}{2}\log_2 s) + 2L - 3 + \log_2 3 \right) s =$$
$$= \left(\frac{L-1}{2}\log_2 k + 2L - 3 + \log_2 3 \right) s <$$
$$< \left(\frac{\log_s k \log_2 k}{2} + 2\log_s k - 1 + \log_2 3 \right) s.$$

Плюс возможен проход по одному переборному графу, требующий не более $2s$ операций. Подставляя дальше, получаем $T(U_I(s)) \leqslant \log_s k(\log_2(5s)+2s-1)+\left(\frac{\log_s k \log_2 k}{2} + 2\log_s k - 1 + \log_2 3\right) s+2s$. В итоге имеем:

$$T(U_I(s)) \leqslant \frac{s}{2}\log_s k \log_2 k + 4s\log_s k + \log_2 k + \\ + (\log_2 5 - 1)\log_s k + (1+\log_2 3)s. \quad (3)$$

Утверждение пункта 1 теоремы 1 следует из оценок (2),(3) для ВИГ вида $U_I(s) \in \mathcal{U}(I, \mathcal{F}_{dom2})$ и того, что $\log_s k = \frac{\log_2 k}{\log_2 s}$.

В таблице 1 показаны асимптотические верхние оценки для $T(U_I(s))$ и $Q(U_I(s))$ в случаях, когда s асимптотически равен значениям, указанным в соответствующем столбце. Параметр α, используемый в качестве показателя степени, считается фиксированным. Таблица демонстрирует, что, варьируя s, можно отдавать предпочтение либо сложности алгоритма, либо объёму требуемой для него памяти.

Таблица 1. Асимптотические верхние оценки решения $U_I(s)$ при $k \to \infty$.

s	$T(U_I(s))$	$Q(U_I(s))$
2	$\log_2^2 k$	$2k\log_2 k$
$\log_2^\alpha k$, $\alpha > 0$	$\frac{\log_2^{2+\alpha} k}{2\alpha \log_2 \log_2 k}$	$\frac{2k\log_2 k}{\alpha \log_2 \log_2 k}$
k^α, $0 < \alpha < 1$	$\frac{k^\alpha \log_2 k}{2\alpha}$	$(8+2/\alpha)k$

5.2. Локальное применение частичного каскадирования

Применим технику частичного каскадирования в каждой внутренней вершине T_V^s. Покажем, как именно это можно сделать, построив ВИГ $U_I'(s,g')$, зависящий не только от параметра s, но и от параметра g', который мы определим ниже.

Как и в предыдущем пункте, строить $U_I'(s,g')$ будем, используя T_V^s в качестве некоего каркаса. Как и раньше, для каждой вершины w_α дерева T_V^s заводим в $U_I'(s,g')$ соответствующую ей вершину v_α. Концевые вершины дерева T_V^s обрабатываются так же, как и раньше. Внутренние же вершины обрабатываются следующим образом. Для каждой внутренней вершины w_α дерева T_V^s строим граф каталогов $L = (G, \mathcal{C})$ в виде цепочки из s элементов, где i-му элементу цепочки соответствует каталог C_i, содержащий упорядоченный набор вторых координат элементов из $V_{\alpha i}$. Здесь $\mathcal{C} = \{C_i\}_{i=1}^s$. Степень графа каталогов L равна $d'(s) = \begin{cases} 1, & \text{если } s = 2; \\ 2, & \text{иначе.} \end{cases}$ Пусть задано нечётное натуральное число $g' \geqslant 4d'(s) + 1$ (то есть $g' \geqslant 5$ при $s = 2$ и $g' \geqslant 9$ для $s \geqslant 3$). Согласно лемме о частичном каскадировании, существует система мостов $S = (G, \mathcal{A}, \mathcal{B})$ ($\mathcal{A} = \{A_i\}_{i=1}^s$), согласованная с графом каталогов L, такая что выполнены следующие условия:

(а) Ширина системы мостов S не превосходит параметра g';

(б) Каждый элемент любого из A_i либо равен значению некоторого элемента из C_j для некоторого j, либо равен $\pm\infty$;

(в) Имеет место неравенство

$$\sum_{i=1}^{s} |A_i| \leqslant (1 + c') \sum_{i=1}^{s} |C_i| + 2s + 4(s-1)c', \tag{4}$$

где $c' = \frac{4d'(s)}{g'+1-4d'(s)}$, а запись $U(C)$ обозначает каталог, получаемый из каталога C путём выбрасывания из него всех дубликатов.

(г) Любой каталог A_i содержит не более одной записи с некоторым значением $x \in \overline{\mathbb{R}}$.

Как и в предыдущем варианте из вершины v_α выпустим такое же дихотомическое дерево с $2s+1$ вершинами $\gamma_0^\alpha, \gamma_1^\alpha, \ldots, \gamma_{2s}^\alpha$. Для вершин с нечётными номерами всё сделаем как и в предыдущем варианте: из вершины γ_{2i-1}^α выпустим предикатное ребро с единичным предикатом в вершину $v_{\alpha i}$ и если $i > 1$, то выпускаем предикатное ребро с тождественно единичным предикатом в соседнюю вершину с чётным номером γ_{2i-2}^α.

Обозначим A_i' каталог, получаемый из A_i путём удаления самого последнего элемента со значением ∞ — последний элемент обязательно имеет такое значение, поскольку все отрезки покрытия всех мостов из $\mathcal{B}$ равны $[-\infty, \infty]$. Ясно, что $|A_i'| = |A_i| - 1$.

Так как A_i может содержать только значения из V_α и $\pm\infty$, то $|A_i| \leqslant |V_\alpha| + 2$. Если вершина α находится на расстоянии l от корня T_V^s, то получаем $|A_i| \leqslant \lceil \frac{k}{s^l} \rceil + 2$.

Обработка вершины с ненулевым чётным номером γ_{2i}^α будет производиться следующим образом.

Для каждой такой вершины рассмотрим соответствующий ей каталог $A_i' = \{a_j^i\}_{j=1}^{n_i}$, где $n_i = |A_i'|$. По этому каталогу строим соответствующее ему множество предикатных вершин $\omega_1^{\alpha,i}, \ldots, \omega_{n_i}^{\alpha,i}$. Элементу a_j^i каталога A_i' при этом соответствует вершина $\omega_j^{\alpha,i}$. Затем для каждой вершины $\omega_j^{\alpha,i}$ найдём множество записей из библиотеки $V_{\alpha i}^j = \{(y_1, y_2, y_*) \in V_{\alpha i} \colon y_2 \leqslant a_j^i\}$. Если $|V_{\alpha i}^j| = N \neq 0$, то обозначим $V_{\alpha i}^j = \{(y_1^1, y_2^1, y_*^1), \ldots, (y_1^N, y_2^N, y_*^N)\}$ и припишем вершине $\omega_j^{\alpha,i}$ функцию изменения состояния $h_z \in H_{\mathcal{Z}}$, где $z = y_*^1 \bigoplus \cdots \bigoplus y_*^N$.

Используя вершину γ_{2i}^α в качестве корня, строим дихотомическое дерево по 2 координате, основанное на вершинах $\omega_1^{\alpha,i}, \ldots, \omega_{n_i}^{\alpha,i}$ со значениями $a_2^i, \ldots, a_{n_i}^i$.

Далее, если $i < s$, то разобьём каталог A_i' на столько последовательных частей, сколько ярусов в мосте между A_i и A_{i+1}. Каждая часть содержит, во-первых, все элементы одного яруса, лежащие в каталоге A_i, и во-вторых, элемент каталога A_i, лежащий на нижнем переходе, ограничивающем этот ярус. Каждая последовательная часть, соответствующая ярусу указанного моста, рассматривается отдельно. Пусть элементы некоторой одной части, соответствующей ярусу с номером u, имеют вид $a^i_r, a^i_{r+1}, \ldots, a^i_t$, где $1 \leqslant r \leqslant t \leqslant n_i$. Если $r = t$, то вершиной $\phi^{\alpha,i}_u$ будем называть вершину $\omega^{\alpha,i}_r$. В противном случае введём новую вершину $\phi^{\alpha,i}_u$ и построим дихотомическое дерево по 2 координате, основанное на вершинах $\omega^{\alpha,i}_r, \ldots, \omega^{\alpha,i}_t$ со значениями $a^i_{r+1}, \ldots, a^i_t$ и корнем $\phi^{\alpha,i}_u$.

Теперь, если $i > 1$, то для каждой из вершин $\omega^{\alpha,i}_j$ делаем следующее. Берём соответствующий ей элемент a^i_j из каталога A'_i (а значит и из A_i, поскольку он содержит A'_i и ещё один элемент со значением ∞) и в мосте $B \in \mathcal{B}$, соединяющем каталоги A_{i-1} и A_i находим ярус, в котором расположен элемент a^i_j (либо ярус, на нижнем переходе которого лежит a^i_j). Пусть этот ярус имеет номер u. Тогда из вершины $\omega^{\alpha,i}_j$ выпустим предикатное ребро с тождественно единичным предикатом в вершину $\phi^{\alpha,i-1}_u$.

Построенный таким образом граф и назовём $U'_I(s, g')$.

Подсчитаем его объём и сложность в худшем случае.

Объём складывается из следующих частей:

- Подграфы переборных алгоритмов, выпускаемые из вершин соответствующих концевым вершинам T^s_V, дают как и в прошлом случае $2k$ рёбер;
- Для каждой внутренней вершины $w_\alpha \in T^s_V$ строилось дихотомическое дерево с $2s+1$ концевыми вершинами и, следовательно, количество ребер в нём равно $4s$. Из вершин с нечётными номерами γ^α_{2i-1} с $2 \leqslant i \leqslant s$ выпускали по два ребра и из вершины γ^α_1 выпускали одно ребро, что даёт ещё $2s-1$ рёбер в сумме. Вместе это будет по $(6s-1)$ рёбер на внутреннюю вершину T^s_V.
- Для каждой внутренней вершины $w_\alpha \in T^s_V$ с использованием техники частичного каскадирования строили структуру данных, содержащую:
  - Для каждого i от 1 до s дихотомическое дерево с $|A'_i| = |A_i|–1$ концевыми вершинами. В таком дереве ровно $2|A_i|–4$ ребра.
  - Для каждого i от 1 до $s-1$ каталог $|A'_i|$ бился на несколько частей и над каждой частью строилось своё дихотомиче-

ское дерево (точнее, над набором вершин, соответствующих элементам этой части). На каждую такую часть из l элементов приходится количество рёбер равное $2l-2$, поэтому самое большее здесь добавится $2|A_i'|-2=2|A_i|-4$ ребра.

– Наконец, для каждого i от 2 до s из вершин, соответствующих элементам A_i', выпускалось по одному ребру, то есть в сумме $|A_i'|=|A_i|-1$ рёбер.

Получаем $\sum_{i=1}^{s}(2|A_i|-4)+\sum_{i=1}^{s-1}(2|A_i|-4)+\sum_{i=2}^{s}(|A_i|-1)\leqslant \sum_{i=1}^{s}(2|A_i|-4)+\sum_{i=1}^{s}(2|A_i|-4)+\sum_{i=1}^{s}(|A_i|-1)\leqslant 5\sum_{i=1}^{s}|A_i|-9s\overset{(4)}{\leqslant} 5(1+c')\sum_{i=1}^{s}|C_i|+20c'(s-1)+s$, где $c'=\frac{4d'(s)}{g'+1-4d'(s)}$. По построению имеем $|C_i|=|V_{\alpha i}|$, при этом для каждой вершины T_V^s, кроме корневой, соответствующее ей множество ровно один раз используется для построения некоторого каталога типа C_i. Поэтому сумма значений вида $\sum_{i=1}^{s}|C_i|$ по всем внутренним вершинам T_V^s не превосходит $(L-1)k$. Пользуясь тем, что количество внутренних вершин T_V^s не более $\frac{k-1}{s-1}$, получаем в итоге не более $5(1+c')(L-1)k+(20c'(s-1)+s)\frac{k-1}{s-1}$ рёбер.

В сумме получаем $Q(U_I'(s,g'))\leqslant 2k+(20c'(s-1)+7s-1)\frac{k-1}{s-1}+5(1+c')(L-1)k$ и итоговая оценка на требуемую память

$$Q(U_I'(s,g'))\leqslant 5(1+c')k\log_s k+(9+20c')k+6\frac{k-1}{s-1}. \qquad (5)$$

Оценим сложность $U_I'(s,g')$ в худшем случае. Оценим сверху количество вычисленных операций: для каждого из $L-1$ уровней T_V^s от 0 до $L-2$ как и в предыдущем варианте производится проход по дихотомическому дереву с $2s+1$ концевыми вершинами, что даёт нам не более $(L-1)\lceil\log_2(2s+1)\rceil\leqslant(L-1)\lceil\log_2(\frac{5s}{2})\rceil\leqslant(L-1)\log_2(5s)\leqslant \log_s k\log_2(5s)\leqslant\log_2 k+\log_2 5\log_s k$ операций. Как и в прошлом варианте, возможен проход по одному переборному графу, требующий не более $2s$ операций. Наконец, для уровня с номером l от 0 до $L-2$ включительно возможны в худшем случае следующие операции:

- первичное прохождение по одному дихотомическому дереву с не более чем $\max_i |A'_i| \leqslant \max_i |A_i| - 1 \leqslant \lceil \frac{k}{s^l} \rceil + 1$ концевыми вершинами;
- не более $s-1$ прохождений по тождественно единичному предикату в маленькие дихотомические деревья с не более чем $g'+1$ концевыми вершинами у каждого, что даёт максимум $(s-1)(1+\lceil \log_2(g'+1) \rceil)$ операций;
- прохождение в вершину вида γ^{α}_{2i-1}, которое приводит к вычислению не более двух тождественных предикатов. Но в таком случае в предыдущем пункте максимальное количество переходов было бы не $s-1$, а уже $s-2$. И так как $1 + \lceil \log_2(g'+1) \rceil$ точно не меньше 2, то два последних пункта вместе дают не более $(s-1)(1+\lceil \log_2(g'+1) \rceil)$ операций.

Итого для уровня l выходит операций не более чем $\lceil \log_2(\lceil \frac{k}{s^l} \rceil + 1) \rceil + (s-1)(1+\lceil \log_2(g'+1) \rceil) \leqslant \log_2(\frac{k}{s^l}+2) + 1 + (s-1)(1+\lceil \log_2(g'+1) \rceil)$. При $l = L-2$ пользуясь тем, что $k > s^{L-1}$ и $s \geqslant 2$, получаем $\log_2(\frac{k}{s^l}+2) \leqslant \log_2(\frac{2k}{s^l})$. При $l < L-2$ имеем $\log_2(\frac{k}{s^l}+2) \leqslant \log_2(\frac{3k}{2s^l})$. Суммарно для всех уровней от 0 до $L-2$ получаем $(L-1)\big(1+(s-1)(1+\lceil \log_2(g'+1) \rceil)\big) + \log_2 2 + (L-2)\log_2 \frac{3}{2} + \sum_{l=0}^{L-2} \log_2 \frac{k}{s^l} \leqslant (L-1)\big(\log_2 k + \log_2 \frac{3}{2} + 1 + (s-1)(1+\lceil \log_2(g'+1) \rceil)\big) + 1 - \log_2 \frac{3}{2} - \frac{(L-2)(L-1)}{2}\log_2 s \leqslant (L-1)\big(\log_2 k + \log_2 3 + (s-1)(1+\lceil \log_2(g'+1) \rceil) - \frac{L}{2}\log_2 s + \log_2 s\big) + \log_2 \frac{4}{3} \leqslant \log_s k\big(\log_2 k + \log_2 3 + (s-1)(1+\lceil \log_2(g'+1) \rceil) - \frac{1}{2}\log_s k \log_2 s + \log_2 s\big) + 1 \leqslant \log_s k\big(\frac{1}{2}\log_2 k + \log_2 3 + (s-1)(1+\lceil \log_2(g'+1) \rceil) + \log_2 s\big) + 1$ операций.

В сумме для всего графа $U'_I(s, g')$ получаем:

$$
\begin{aligned}
T(U'_I(s,g')) &\leqslant \log_2 k + \log_2 5 \log_s k + 2s + \\
&+ \log_s k \left(\frac{1}{2}\log_2 k + \log_2 3 + (s-1)(1+\lceil \log_2(g'+1) \rceil) + \log_2 s\right) + 1 \leqslant \\
&\leqslant \log_s k \left(\frac{1}{2}\log_2 k + \log_2 15 + (s-1)(1+\lceil \log_2(g'+1) \rceil)\right) + \\
&+ 2\log_2 k + 2s + 1. \qquad (6)
\end{aligned}
$$

Утверждение пункта 2 теоремы 1 следует из оценок (5),(6) для ВИГ вида $U'_I(s,g') \in \mathcal{U}(I, \mathcal{F}_{dom2})$ и того, что $\log_s k = \frac{\log_2 k}{\log_2 s}$.

В таблице 2 показаны асимптотические верхние оценки для $T(U'_I(s,g'))$ и $Q(U'_I(s,g'))$ в случаях, когда s асимптотически равен значениям, указанным в соответствующем столбце. Параметр g' считается фиксированным нечётным натуральным числом не менее 5 для $s = 2$ и не менее 9 для остальных случаев. Параметр α, используемый в качестве показателя степени, также считается фиксированным.

Таблица 2. Асимптотические верхние оценки решения $U'_I(s,g')$ при $k \to \infty$.

s	$T(U'_I(s,g'))$	$Q(U'_I(s,g'))$
2	$\frac{1}{2}\log_2^2 k$	$5\frac{g'+1}{g'-3} \cdot k\log_2 k$
$\log_2 k$	$\frac{(\frac{3}{2}+\lceil\log_2(g'+1)\rceil)\log_2^2 k}{\log_2\log_2 k}$	$5\frac{g'+1}{g'-7} \cdot \frac{k\log_2 k}{\log_2\log_2 k}$
$\log_2^\alpha k$, $\alpha \in (0,1)$	$\frac{\log_2^2 k}{2\alpha\log_2\log_2 k}$	$5\frac{g'+1}{g'-7} \cdot \frac{k\log_2 k}{\alpha\log_2\log_2 k}$
$\log_2^\alpha k$, $\alpha > 1$	$\frac{(1+\lceil\log_2(g'+1)\rceil)\log_2^{1+\alpha} k}{\alpha\log_2\log_2 k}$	$5\frac{g'+1}{g'-7} \cdot \frac{k\log_2 k}{\alpha\log_2\log_2 k}$
k^α, $0 < \alpha < 1$	$k^\alpha\left(\frac{1+\lceil\log_2(g'+1)\rceil}{\alpha}+2\right)$	$(9+\frac{5}{\alpha}+(160+\frac{40}{\alpha})\frac{1}{g'-7})k$

5.3. Глобальное применение частичного каскадирования

В отличие от предыдущего варианта, когда частичное каскадирование применялось по отдельности для каждой внутренней вершины T_V^s, теперь мы будем применять частичное каскадирование один раз. Будем строить ВИГ $U''_I(s,g'')$, зависящий от параметра s и вводимого ниже параметра g''.

Построим граф каталогов $P_V^s = (\mathcal{P}, \mathcal{C})$. Будем обозначать вершины графа $\mathcal{P}$ как множество P, а его рёбра — E, то есть $\mathcal{P} = (P, E)$. Здесь $\mathcal{C} = \{C_\pi\}_{\pi \in P}$ — каталоги, соответствующие вершинам $\mathcal{P}$. Строить P_V^s будем следующим образом: для каждой вершины $w_\alpha \in T_V^s$ заведём соответствующую ей вершину π_α в строящемся графе каталогов P_V^s. Для каждой внутренней вершины $w_\alpha \in T_V^s$ из π_α выпускаем бинарное дерево минимально возможной высоты с $2s+1$ концевыми вершинами и такое, что каждая внутренняя вершина имеет ровно двух потомков (всё это аналогично построению дихотомического дерева). Концевые вершины этого дерева мы назовём $\beta_0^\alpha, \ldots, \beta_{2s}^\alpha$. Левая вершина β_0^α нам не потребуется, и мы тут же убираем её вместе с ведущим в неё ребром. В дальнейшем мы несколько раз будем упоминать о вершинах вида β_0^α, называя их также *удалёнными* вершинами. Главное что эти вершины не входят в граф P_V^s, к которому мы будем применять следствие из теоремы о частичном каскадировании. Далее каждую вершину с нечётным номером β_{2i-1}^α соединяем ребром с вершиной $\pi_{\alpha i}$ (это та вершина, которая в P_V^s соответствует вершине $w_{\alpha i} \in T_V^s$). Если при этом $i > 1$, то соединяем β_{2i-1}^α ещё и с вершиной β_{2i-2}^α. Вершины с чётными номерами начиная с 2 соединяем по очереди в цепочку, то есть проводим рёбра $\{\beta_2^\alpha, \beta_4^\alpha\}, \ldots, \{\beta_{2s-2}^\alpha, \beta_{2s}^\alpha\}$. Каждой вершине вида $\pi = \beta_{2i}^\alpha \in P$ ставим в соответствие каталог C_π, состоящий из упорядоченных по возрастанию вторых координат записей из подбазы $V_{\alpha i}$. Всем остальным вершинам P ставим в соответствие пустые каталоги. На этом построение графа каталогов завершается.

Несложно убедиться, что степень построенного графа каталогов P_V^s равна $d''(s) = \begin{cases} 3, & \text{если } s = 2; \\ 4, & \text{иначе.} \end{cases}$ Пусть задано нечётное натуральное число $g'' \geqslant 4d''(s) + 1$ (то есть $g' \geqslant 13$ при $s = 2$ и $g' \geqslant 17$ для $s \geqslant 3$). Согласно лемме о частичном каскадировании, существует система мостов $S = (\mathcal{P}, \mathcal{A}, \mathcal{B})$ (где $\mathcal{A} = \{A_\pi\}_{\pi \in P}$), согласованная с графом каталогов P_V^s, такая что выполнены следующие условия:

(а) Ширина системы мостов S не превосходит параметра g'';

(б) Каждый элемент любого из A_π либо равен значению некоторого элемента из C_v для некоторого $v \in P$, либо равен $\pm\infty$;

(в) Имеет место следующее неравенство:

$$\sum_{\pi \in P} |A_\pi| \leqslant (1 + c'') \sum_{\pi \in P} |C_\pi| + 2|P| + 4c''|E|\,, \tag{7}$$

где $c'' = \frac{4d''(s)}{g''+1-4d''(s)}$, а запись $U(C)$ обозначает каталог, получаемый из каталога C путём выбрасывания из него всех дубликатов.

(г) Любой каталог A_π содержит не более одной записи с некоторым значением $x \in \overline{\mathbb{R}}$.

Используя тот факт, что для каждой записи из V её вторая координата используется для построения не более чем $L-1$ каталога типа C_π, получаем:

$$\sum_{\pi \in P} |C_\pi| \leqslant (L-1)k < k \log_s k. \tag{8}$$

Посчитаем величины $|P|$ и $|E|$. Из построения графа каталогов P_V^s видно, что если убрать из него все рёбра, соединяющие вершины вида β_i^α между собой, то получится дерево. Количество убранных рёбер равно $2(s-1)t_{in}$: для каждой внутренней вершины $w_\alpha \in T_V^s$ строилось $s-1$ рёбер, связывающих между собой вершины β_i^α с чётными номерами i, и $s-1$ рёбер, связывающих вершины с нечётными номерами и лежащие левее них вершины с чётными номерами. Используя тот факт, что в любом дереве количество рёбер на единицу меньше количества вершин, получаем что $|E| = |P| - 1 + 2(s-1)t_{in}$. Оценим количество вершин $|P|$: для каждой концевой вершины дерева T_V^s мы добавляли одну вершину в P, а для каждой внутренней вершины T_V^s — целое бинарное дерево с $2s+1$ концевыми вершинами, но потом одну вершину исключали. На одну внутреннюю вершину T_V^s мы таким образом добавляли $2(2s+1)-1-1 = 4s$ вершин в P. В итоге имеем:

$$|P| = 4s\, t_{in} + t_{leaf} \overset{(1)}{\leqslant} 4s\, \frac{k-1}{s-1} + k = 5k - 4 + 4\frac{k-1}{s-1}\,, \tag{9}$$

$$|E| = (6s-2)t_{in} - 1 + t_{leaf} \overset{(1)}{\leqslant} (6s-2)\frac{k-1}{s-1} - 1 + k = 7k - 7 + 4\frac{k-1}{s-1}. \tag{10}$$

Для каждой вершины π графа P_V^s каталог A_π обязательно содержит элементы со значениями $\pm\infty$. Обозначим каталог, получающийся из A_π путём удаления последней записи как A'_π. Удалённая запись при этом всегда будет иметь значение ∞. Оценим сверху суммарный размер всех A'_π:

$$\sum_{\pi\in P}|A'_\pi| = \sum_{\pi\in P}|A_\pi| - |P| \overset{(7)}{\leqslant} (1+c'')\sum_{\pi\in P}|C_\pi| + |P| + 4c''|E| \overset{(8)}{\leqslant}$$
$$\overset{(8)}{\leqslant} (1+c'')k\log_s k + |P| + 4c''|E| \quad (11)$$

Построим по графу P^s_V и согласованной с ним системе мостов $S = (\mathcal{P},\mathcal{A},\mathcal{B})$ ВИГ $U''_I(s,g'')$, решающий задачу I. Рассмотрим сперва граф каталогов P^s_V. Его рёбра не являются ориентированными, но для дальнейших рассуждений нам будет удобно задать для них ориентацию следующим образом: для вершины $\pi_\alpha \in P$ все рёбра бинарного дерева, выходящего из π_α направлены от корня; рёбра вида $\{\beta^\alpha_{2i-1},\beta^\alpha_{2i-2}\}$ и $\{\beta^\alpha_{2i},\beta^\alpha_{2i-2}\}$ направлены от вершины с большим индексом к вершине с меньшим индексом; наконец, рёбра вида $\{\beta^\alpha_{2i-1},\pi_{\alpha i}\}$ ведут из β^α_{2i-1} в $\pi_{\alpha i}$.

Начнём собственно построение $U''_I(s,g'')$. Для каждой вершины $\pi\in P$ рассмотрим каталог A'_π. Пусть $|A'_\pi| = n_\pi$ и $A'_\pi = \{a^\pi_1,\ldots,a^\pi_{n_\pi}\}$. Для каждого элемента a^π_j при $1\leqslant j\leqslant n_\pi$ заведём в $U''_I(s,g'')$ соответствующую ему вершину θ^π_j. Из каждой вершины $\pi_\alpha\in P$, соответствующей концевой вершине w_α дерева T^s_V выпустим граф перебора $U^e_{V_\alpha}$.

Затем, рассмотрим все рёбра графа P^s_V. Пусть некоторое ребро e ведёт из вершины v в вершину π. Разобьём каталог A'_π на столько последовательных частей, сколько ярусов в мосте между A_v и A_π. Каждая часть содержит, во-первых, все элементы одного яруса, лежащие в каталоге A_π, и во-вторых, элемент каталога A_π, лежащий на нижнем переходе, ограничивающем этот ярус. Каждая последовательная часть, соответствующая ярусу указанного моста, рассматривается отдельно. Пусть элементы некоторой одной части, соответствующей ярусу с номером u, имеют вид $a^\pi_r, a^\pi_{r+1},\ldots,a^\pi_t$, где $1\leqslant r\leqslant t\leqslant n_\pi$. Если $r=t$, то вершиной η^e_u будем называть вершину θ^π_r. В противном случае введём новую вершину η^e_u и построим дихотомическое дерево по 2 координате, основанное на вершинах $\theta^\pi_r,\ldots,\theta^\pi_t$ со значениями $a^\pi_{r+1},\ldots,a^\pi_t$ и корнем в η^e_u.

Затем для каждой вершины θ_j^v берём соответствующий ей элемент a_j^v из каталога A'_v (а значит и из A_v, поскольку он содержит A'_v и ещё один элемент со значением ∞) и в мосте $B \in \mathcal{B}$, соединяющем каталоги A_v и A_π находим ярус, в котором расположен элемент a_j^v (либо ярус, на нижнем переходе которого лежит a_j^v). Пусть этот ярус имеет номер u. Тогда *точкой перехода из вершины* θ_j^v *по ребру* e назовём вершину η_u^e.

Рассмотрим корневую вершину $\pi = \pi_\lambda$ графа каталогов P_V^s. Ей в $U''_I(s, g'')$ соответствуют вершины $\theta_1^\pi, \ldots, \theta_{n_\pi}^\pi$. Если $n_\pi = 1$, то вершину θ_1^π объявим корнем $U''_I(s, g'')$, иначе заведём новую вершину, которую объявим корнем $U''_I(s, g'')$ и, используя её как корень, построим дихотомическое дерево по 2 координате, основанное на вершинах $\theta_1^\pi, \ldots, \theta_{n_\pi}^\pi$ со значениями $a_2^\pi, \ldots, a_{n_\pi}^\pi$. Заметим, что количество концевых вершин построенного дерева не превосходит $k+1$, поскольку каталог A'_π может содержать только элементы со значениями либо равными второй координате некоторой записи библиотеки, либо равными $-\infty$.

Добавим к графу $U''_I(s, g'')$ ещё одну вершину δ, которая будет использоваться как тупиковая.

К данному моменту мы уже объявили все вершины, которые будут присутствовать в графе $U''_I(s, g'')$. Продолжим построение требуемого графа. Для каждой внутренней вершины w_α дерева T_V^s в графе каталогов имеется соответствующая вершина π_α, из которой выходит бинарное дерево с концевыми вершинами $\beta_0^\alpha, \ldots, \beta_{2s}^\alpha$ (причём вершина β_0^α потом удалялась из P_V^s). Для каждой внутренней вершины π этого дерева мы уже завели целый набор вершин $U''_I(s, g'')$: $\theta_1^\pi, \ldots, \theta_{n_\pi}^\pi$. Все эти вершины объявляются переключательными и каждой из них приписывается один и тот же переключатель $g^2_{a_p^\pi}$, где p — наименьший из таких индексов, что β_p^α находится в правом поддереве относительно π. Из вершины π либо выходит два ребра e_{left} и e_{right} в левое и правое поддеревья, либо только одно ребро e_{right}, если левое ребро вело в удалённую вершину β_0^α (будем говорить тогда, что e_{left} не определено). Из каждой вершины θ_j^π с $1 \leqslant j \leqslant n_\pi$ выпускаем по два ребра. Если e_{left} не определено, то первое ребро выпустим в тупиковую вершину δ, в противном случае первое ребро будет вести в точку перехода из вершины θ_j^π по ребру e_{left}. Припишем первому ребру число 1. Второе ребро из θ_j^π выпустим в точку перехода из вершины θ_j^π по ребру e_{right}. Второму ребру припишем число 2.

Теперь рассмотрим вершины с нечётными индексами $\beta_1^\alpha, \ldots, \beta_{2s-1}^\alpha$. Для каждой такой вершины $\pi = \beta_{2i-1}^\alpha$ $(1 \leqslant i \leqslant s)$ мы заводили набор вершин в графе $U_I''(s, g'')$: $\theta_1^\pi, \ldots, \theta_{n_\pi}^\pi$. Каждую такую вершину θ_j^π объявляем предикатной и выпускаем из неё ребро с тождественно единичным предикатом в точку перехода из вершины θ_j^π по ребру, соединяющему $\pi = \beta_{2i-1}^\alpha$ с $\pi_{\alpha i}$. Если же при этом $i > 1$, то из каждой θ_j^π выпустим также ребро с тождественно единичным предикатом в точку перехода из θ_j^π по ребру, соединяющему $\pi = \beta_{2i-1}^\alpha$ и β_{2i-2}^α.

Затем рассмотрим вершины с нечётными индексами $\beta_2^\alpha, \ldots, \beta_{2s}^\alpha$. Для каждой такой вершины $\pi = \beta_{2i}^\alpha$ $(1 \leqslant i \leqslant s)$ мы тоже заводили набор вершин в графе $U_I''(s, g'')$: $\theta_1^\pi, \ldots, \theta_{n_\pi}^\pi$. Рассмотрим каждую из этих вершин θ_j^π. Эта вершина соответствует значению a_j^π каталога A'_π. Положим множество $V_j^\pi = \{(y_1, y_2, y_*) \in V_{\alpha i} \colon y_2 \leqslant a_j^\pi\}$. Если $V_j^\pi \neq \varnothing$, то припишем вершине θ_j^π функцию изменения состояния $h_z \in H_{\mathcal{Z}}$, где $z = \bigoplus\limits_{(y_1, y_2, y_*) \in V_j^\pi} y_*$. Объявим вершину θ_j^π предикатной и если $i > 1$, то выпустим из неё ребро с тождественно единичным предикатом в точку перехода из θ_j^π по ребру, соединяющему $\pi = \beta_{2i}^\alpha$ и β_{2i-2}^α.

Построение графа $U_I''(s, g'')$ на этом завершается. Оценим его объём и сложность в худшем случае.

Начнём с объёма $Q(U_I''(s, g''))$. Как и в предыдущих вариантах, $2k$ рёбер добавляют подграфы переборных алгоритмов. При объявлении корня $U_I''(s, g'')$ мы строили бинарное дерево с не более чем $k+1$ концевыми вершинами. Количество добавленных рёбер при этом не превосходит $2(k+1) - 2 = 2k$.

Для каждой вершины $\pi \in P_V^s$ и каждого ребра $e \in E$, входящего в π мы строили вершины, являющиеся точками входа по ребру e. При этом последовательность вершин $\theta_1^\pi, \ldots, \theta_{n_\pi}^\pi$ разбивалась на одну или несколько частей и каждая часть использовалась как множество концевых вершин для некоторого бинарного дерева. При фиксированных π и e эти деревья в сумме содержат не более $2|A'_\pi| - 2$ ребра.

Далее, для каждой вершины $\pi \in P_V^s$ и каждого ребра $e \in E$, исходящего из π мы выпускали из каждой вершины θ_j^π по одному ребру, что при фиксированных π и e даёт нам ровно $|A'_\pi|$ рёбер. Плюс для тех вершин $\pi \in P$, из которых выпускались удалённые рёбра в вершины вида β_0^α, из каждой вершины θ_j^π выпускалось также по одному ребру, что тоже даёт ровно $|A'_\pi|$ рёбер.

Для вершины π с количеством входящих и исходящих рёбер равным d_{in} и d_{out} соответственно, это даёт в сумме не более $(2|A'_\pi| - 2)d_{in} + |A'_\pi|d_{out}$ рёбер. Здесь и далее для удобства когда мы будем говорить о количестве исходящих из вершины π рёбер, мы будем считать и удалённые рёбра, ведущие в удалённые вершины вида β_0^α.

При $s = 2$ в графе P_V^s могут встречаться только вершины с $(d_{in}, d_{out}) \in \{(0,2),(1,2),(1,1),(3,0)\}$, поэтому на каждую вершину π добавится не более $\max(4|A'_\pi| - 2,\ 6|A'_\pi| - 6)$ рёбер. Но $|A'_\pi| \geqslant 1$ (поскольку A'_π обязательно содержит элемент со значением $-\infty$). Поэтому $\max(4|A'_\pi| - 2, 6|A'_\pi| - 6) \leqslant 6|A'_\pi| - 4$.

При $s \geqslant 3$ в графе P_V^s могут встречаться только вершины с $(d_{in}, d_{out}) \in \{(0,2),(1,2),(1,1),(3,1),(3,0)\}$, поэтому на каждую вершину π добавится не более $\max(4|A'_\pi| - 2,\ 7|A'_\pi| - 6) \leqslant 7|A'_\pi| - 5$ рёбер.

Итого, при $s = 2$ получаем:

$$
\begin{aligned}
Q(U''_I(s,g'')) &\leqslant 2k + 2k + \sum_{\pi \in P}(6|A'_\pi| - 4) \overset{(11)}{\leqslant} \\
&\overset{(11)}{\leqslant} 4k + 6(1+c'')k\log_s k + 2|P| + 24c''|E| \overset{(9),(10)}{\leqslant} \\
&\overset{(9),(10)}{\leqslant} 6(1+c'')k\log_s k + 14k - 8 + 8\frac{k-1}{s-1} + 24c''\left(7k - 7 + 4\frac{k-1}{s-1}\right) \overset{s=2}{=} \\
&\overset{s=2}{=} 6(1+c'')k\log_2 k + 22k - 16 + 264c''(k-1). \qquad (12)
\end{aligned}
$$

Аналогично, для $s \geqslant 3$:

$$
\begin{aligned}
Q(U''_I(s,g'')) &\leqslant 2k + 2k + \sum_{\pi \in P}(7|A'_\pi| - 5) \overset{(11)}{\leqslant} \\
&\overset{(11)}{\leqslant} 4k + 7(1+c'')k\log_s k + 2|P| + 28c''|E| \overset{(9),(10)}{\leqslant}
\end{aligned}
$$

$$\overset{(9),(10)}{\leqslant} 7(1+c'')k\log_s k+14k-8+8\frac{k-1}{s-1}+28c''\left(7k-7+4\frac{k-1}{s-1}\right). \quad (13)$$

Оценим теперь сложность $T(U''_I(s,g''))$. Заметим, что прохождение по любой точке перехода предполагает спуск по бинарному дереву у которого не более $g''+1$ концевая вершина, поэтому количество операций, вычисляемых при этом не превосходит $\lceil\log_2(g''+1)\rceil$. Ответ на один запрос состоит самое большее в одном прохождении по бинарному дереву с не более чем $k+1$ концевыми вершинами и затем для не более чем $L-1$ уровня: сначала спуск по бинарному дереву с не более чем $2s+1$ концевыми вершинами, причём переход от родителя к потомку здесь занимает не одну, а максимум $1+\lceil\log_2(g''+1)\rceil$ операций. После спуска мы попадаем либо в одну из вершин, соответствующих некоторому β_j^α с чётным номером j, либо с нечётным (либо в тупиковую вершину, но тогда на этом вычисления заканчиваются). Для чётных j может последовать самое большее $s-1$ переход по $1+\lceil\log_2(g''+1)\rceil$ операций, для нечётных может произойти $2+(s-2)$ переход, один из которых ведёт вниз. Таким образом на уровне l после спуска производится не более $(1+\lceil\log_2(g''+1)\rceil)\,s$ операций. Итого на уровне l не более $(s+\lceil\log_2(2s+1)\rceil)\,(1+\lceil\log_2(g''+1)\rceil)$ операций. По всем уровням это будет не более $(L-1)\,(s+\lceil\log_2(2s+1)\rceil)\,(1+\lceil\log_2(g''+1)\rceil)\leqslant$ $\log_s k\,(s+\lceil\log_2(2s+1)\rceil)\,(1+\lceil\log_2(g''+1)\rceil)$ операций. Наконец, возможен проход по одному переборному графу, требующий самое большее $2s$ операций. В итоге, получаем:

$$T(U''_I(s,g''))\leqslant \log_s k\,(s+\lceil\log_2(2s+1)\rceil)\left(1+\lceil\log_2(g''+1)\rceil\right)+2s \quad (14)$$

Утверждения пунктов 3 и 4 теоремы 1 следует из оценок (12), (13), (14) для ВИГ вида $U''_I(s,g'')\in\mathcal{U}(I,\mathcal{F}_{dom2})$ и того, что $\log_s k=\frac{\log_2 k}{\log_2 s}$.

В таблице 3 показаны асимптотические верхние оценки для $T(U''_I(s,g''))$ и $Q(U''_I(s,g''))$ в случаях, когда s асимптотически равен значениям, указанным в соответствующем столбце. Параметр g'' считается фиксированным нечётным натуральным числом не менее 13 для $s=2$ и не менее 17 для остальных случаев. Параметр α, используемый в качестве показателя степени, также считается фиксированным.

Таблица 3. Асимптотические верхние оценки решения $U''_I(s,g'')$ при $k\to\infty$.

s	$T(U''_I(s,g''))$	$Q(U''_I(s,g''))$
2	$5(1+\lceil\log_2(g''+1)\rceil)\log_2 k$	$6\frac{g''+1}{g''-11}\cdot k\log_2 k$
$\log_2^\alpha k,\ \alpha>0$	$(1+\lceil\log_2(g''+1)\rceil)\frac{\log_2^{1+\alpha} k}{\alpha\log_2\log_2 k}$	$7\frac{g''+1}{g''-15}\cdot\frac{k\log_2 k}{\alpha\log_2\log_2 k}$
$k^\alpha,\ 0<\alpha<1$	$(\frac{1+\lceil\log_2(g''+1)\rceil}{\alpha}+2)k^\alpha$	$(\frac{7(g''+1)}{\alpha(g''-15)}+14+\frac{3136}{g''-15})k$

6. Заключение

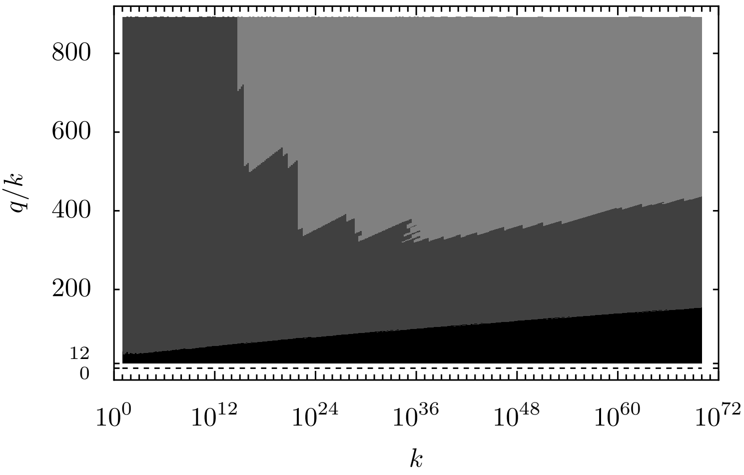

Рис. 2. Карта превосходства семейств алгоритмов поиска. Тёмным выделена область превосходства семейства $U_I(s)$, более светлым — $U'_I(s,g')$ и самым светлым цветом — $U''_I(s,g'')$.

В данной статье мы в рамках информационно-графовой модели данных рассмотрели три параметризованных семейства алгоритмов, решающих задачу подсчёта для двумерной задачи о доминировании. Первое семейство (ВИГ вида $U_I(s)$) является обобщением метода дерева регионов, второе ($U'_I(s,g')$) получается из первого с помощью многократного (каждый раз для небольшой части структуры) применения техники частичного каскадирования и позволяет дополнительно уменьшить время работы алгоритма за счёт некоторого увеличения объёма. Третье семейство $(U_I''(s, g''))$ получается также из первого путём однократного применения техники частичного каскадирования по всей структуре и позволяет в некоторых случаях ещё сильнее уменьшить время работы алгоритма, используя ещё больше памяти.

На рисунке 2 приведена карта, полученная с помощью вычислений на компьютере. По оси абсцисс откладывается количество записей в базе данных k, по оси ординат — отношение доступной памяти q и числа k. При заданных k и q проверялось, какое из трёх семейств алгоритмов позволяет получить самые лучшие оценки на время работы алгоритма для базы из k элементов, при том что объём памяти занимаемой алгоритмом не превосходит q. Если лучший результат удавалось получить с помощью первого семейства, то область помечалась тёмной, если вторым — более светлым, а если третьим, то самым светлым. На рисунке видно, что при любых k до определённого порога доступной памяти q лучшие результаты дают алгоритмы первого семейства. При последующем увеличении q лидерство переходит к алгоритмам второго семейства. Далее ситуация зависит от величины k. Если k больше определённого порога (около 10^{15}), то при дальнейшем росте q лучшие оценки на время начинает показывать третье семейство алгоритмов.

Полученные данные свидетельствуют в пользу того, что все три семейства алгоритмов потенциально имеют право на существование, поскольку каждое из них имеет свои области превосходства. При этом следует отметить, что третье семейство всё же представляет более теоретический, чем практический интерес, поскольку позволяет получить преимущество только для невероятно больших баз данных и ещё большего объёма доступной памяти.

Список литературы

[1] Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. М.: Мир, 1989.

[2] Chazelle B., Guibas L. J. Fractional cascading: I. A Data Structuring Technique // Algorithmica. 1. 1986. P. 133–162.

[3] Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации. М.: ФИЗМАТЛИТ, 2002.

[4] Пивоваров А. П. Моделирование вычислительных задач информационного поиска // Интеллектуальные системы. Т. 14, вып. 1–4. 2010. С. 229–250.

[5] Пивоваров А. П. Техника частичного каскадирования для итеративного поиска в линейно упорядоченных множествах // Интеллектуальные системы. Т. 15, вып. 1–4. 2011. С. 205–222.