Техника частичного каскадирования для итеративного поиска в линейно упорядоченных множествах

А. П. Пивоваров

Итеративный поиск представляет собой поиск одного и того же ключа в нескольких линейно упорядоченных списках. Техника частичного каскадирования [1] позволяет отказаться от отдельного бинарного поиска на каждой итерации. Количество требуемой памяти при этом может возрасти не более чем в константное число раз, однако каждая итерация, кроме первой, может быть произведена за меньшее время, ограниченное параметрами ветвления итеративного поиска. В данной статье описывается техника частичного каскадирования и производится оценка размеров получаемых структур данных.

Ключевые слова: частичное каскадирование, итеративный поиск, компромисс между временем работы и требуемой памятью.

1. Введение

Техника частичного каскадирования, впервые описанная в статье Бернарда Чазелле и Леонидаса Гуибаса [1], представляет собой способ организации данных для решения задачи итеративного поиска. В работе [2] приведён ряд приложений техники частичного каскадирования в геометрических задачах поиска.

Общий вид задачи итеративного поиска описан ниже в соответствующем разделе. Здесь мы лишь в общих чертах опишем эту задачу. Пусть задан граф, каждой вершине v которого поставлен в соответствие упорядоченный список действительных чисел C_v. Запросом является пара (x,π), где x — некоторое действительное число, а π — путь в графе. Задача состоит в том, чтобы для каждой вершины v пути π находить первый элемент в C_v, значение которого больше или равно x.

Идея частичного каскадирования состоит в том, что некоторая доля элементов списка C_v перебрасывается его соседним (по графу) спискам и наоборот. При этом некоторая часть элементов, переброшенных в соседний список, может быть переброшена дальше его соседям и так далее (это и есть то самое частичное каскадирование, которое дало название описываемой технике). Цель этого обмена элементами состоит в том, чтобы у соседних списков с некоторой гарантированной частотой встречались элементы с одинаковыми значениями, благодаря чему между списками можно переходить от ответа для одного списка к ответу для соседнего списка.

В работе [1] исследуются такие аспекты техники частичного каскадирования, как время построения структуры данных, память, занимаемая построенной структурой и время ответа на запрос. Также уделяется внимание возможности динамизации техники частичного каскадирования.

В настоящей работе внимание сосредоточено на статическом варианте. Производится аккуратная оценка тех характеристик получаемых структур, которые не зависят от модели вычислений. Кроме того, параметр, отвечающий за гарантированную частоту появления элементов с одинаковыми значениями в соседних списках, в данной работе берётся произвольным в отличие от [1].

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

2. Основные понятия и формулировка результатов

Сформулируем определения, которые понадобятся нам в дальнейшем.

Под *расширенными действительными числами* будем понимать элементы множества $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

Каталогами будем называть неубывающие последовательности расширенных действительных чисел. Например, последовательность $-\infty, -5, 0, 0, 7, +\infty, +\infty$ — каталог из семи элементов.

Один каталог будем называть *подкаталогом* другого, если он как последовательность является подпоследовательностью второго. Каталог $-5, 7, +\infty, +\infty$ — подкаталог указанного выше каталога.

Пусть C — каталог. Тогда каталог, получающийся из него удалением дубликатов будем обозначать $U(C)$. Например, если $C = -\infty, -5, 0, 0, 7, +\infty, +\infty$, то $U(C) = -\infty, -5, 0, 7, +\infty$.

Пусть C — некоторый каталог, а $x \in \overline{\mathbb{R}}$. Будем обозначать $\xi(x, C)$ первый элемент каталога C, значение которого не меньше, чем x. Если такого элемента нет, будем писать $\xi(x, C) = \Lambda$.

Здесь и далее действительные числа используются для простоты — вместо них можно использовать любое другое линейно упорядоченное множество.

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. Для удобства 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 $i_k < i < i_{k+1}$. Очевидно, каждый элемент каталога C принадлежит не более чем одному ярусу моста B, но может не принадлежать ни одному ярусу если для него выполнено одно из условий:

1) элемент принадлежит одному из переходов моста;
2) элемент лежит выше верхнего перехода моста;
3) элемент 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*.

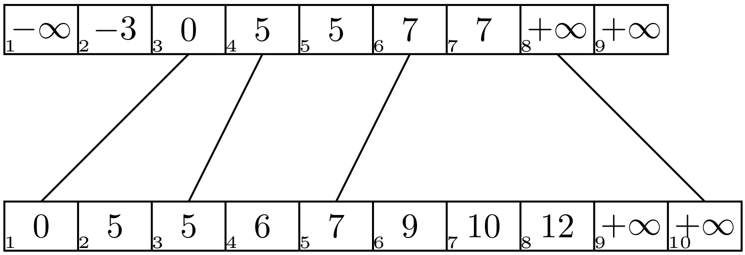

Рис. 1. Пример двух каталогов и моста между ними.

На рисунке 1 приведён пример двух каталогов и моста между ними. Изображённый на рисунке мост состоит из четырёх переходов, поэтому у этого моста три яруса. Первый ярус ширины 1 состоит из одного элемента нижнего каталога с индексом 2 и значением 5. Второй ярус состоит из двух элементов: 5-й элемента верхнего каталога и 4-й элемент нижнего. Третий ярус состоит из пяти элементов: 7-й элемент верхнего каталога и элементы с номерами от 6 до 9 нижнего каталога. Ширина моста определяется шириной самого большого яруса и равна пяти. Отрезок покрытия моста определяется значениями на первом и последнем мостах и равен $[0, +\infty]$.

Графом каталогов будем называть тройку $L = (G, \mathcal{C}, \mathcal{R})$, 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, $\mathcal{R} = \{R_e\}_{e \in E}$ — набор отрезков $R_e = [a_e, b_e]$ ($a_e, b_e \in \overline{\mathbb{R}}$ и $a_e < b_e$), поставленных в соответствие рёбрам G.

Локальной степенью вершины v графа каталогов $L = (G, \mathcal{C}, \mathcal{R})$ будем называть величину $d(v) = \max\limits_{x \in \overline{\mathbb{R}}} |\{e \in E \colon v \in e, x \in [a_e, b_e]\}|$.

Локальной степенью графа каталогов $L = (G, \mathcal{C}, \mathcal{R})$ будем называть величину $d(L) = \max_{v \in V} d(v)$. Как видно из определения, локальная степень графа каталогов зависит только от G и $\mathcal{R}$, но не зависит от $\mathcal{C}$.

Системой мостов будем называть четвёрку $S = (G, \mathcal{A}, \mathcal{R}, \mathcal{B})$, где $(G, \mathcal{A}, \mathcal{R})$ 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 отрезку R_e;
3) для любого $x \in \overline{\mathbb{R}}$ каждый мост из $\mathcal{B}$ содержит не более одного x-перехода.

Систему мостов $S = (G, \mathcal{A}, \mathcal{R}, \mathcal{B})$ будем называть *строгой*, если каждый элемент любого каталога A_v, приписанного любой вершине $v \in V$, либо не принадлежит ни одному переходу ни одного моста из $\mathcal{B}$, либо принадлежит только одному переходу одного моста из $\mathcal{B}$;

Некоторый элемент каталога A_v будем называть *свободным* в том случае, если он не принадлежит ни одному переходу ни одного моста из $\mathcal{B}$.

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

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

Теорема 1. *Пусть имеется граф каталогов* $L = (G, \mathcal{C}, \mathcal{R})$, *где* $G = (V, E)$ *и* $|V| = n$, $|E| = m$. *Пусть* $d = d(L)$ *— локальная степень графа каталогов* L. *Пусть зафиксирован нечётный натуральный параметр* $g_0 \geqslant 4d + 1$. *Тогда существует строгая система мостов* $S = (G, \mathcal{A}, \mathcal{R}, \mathcal{B})$, *согласованная с графом каталогов* L *и при этом выполнены следующие условия:*

(a) *ширина системы мостов* S *не превосходит параметра* g_0;

(б) *имеет место неравенство* $\sum_{v \in V} |A_v| \leqslant (1+c)(\sum_{v \in V} |U(C_v)| + 4m)$*, где* $c = c(g_0) = \frac{4d}{g_0+1-4d}$*;*

(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 нижнему или верхнему концу некоторого отрезка* $R \in \mathcal{R}$*;*

(г) *суммарное количество свободных элементов во всех каталогах* $\mathcal{A}$ *в точности равно* $\sum_{v \in V} |U(C_v)|$*.*

Следствие 1. *Пусть выполнены условия теоремы 1. Тогда существует система мостов* $S = (G, \mathcal{A}, \mathcal{R}, \mathcal{B})$ *(уже не обязательно строгая), согласованная с графом каталогов* L *и при этом выполнены следующие условия:*

(а) *ширина системы мостов* S *не превосходит параметра* g_0*;*

(б) *имеет место неравенство* $\sum_{v \in V} |A_v| \leqslant (1+c)(\sum_{v \in V} |U(C_v)| + 4m)$*, где* $c = c(g_0) = \frac{4d}{g_0+1-4d}$*;*

(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 нижнему или верхнему концу некоторого отрезка* $R \in \mathcal{R}$*;*

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

Если к тому же все отрезки $R_e \in \mathcal{R}$ *равны между собой, то имеет место слегка изменённое ограничение на суммарное количество записей во всех каталогах:*

$$\sum_{v \in V} |A_v| \leqslant (1+c) \sum_{v \in V} |U(C_v)| + 2n + 4mc.$$

3. Общий вид задачи итеративного поиска

Дадим определение задачи итеративного поиска. Мы не будем полностью формализовывать здесь задачу и подход к её решению.

Основной задачей статьи является обсчёт тех параметров получаемых структур, которые не зависят от модели, в которой рассматривается задача поиска.

Пусть задан граф каталогов $L = (G, \mathcal{C}, \mathcal{R})$, где $G = (V, E)$ — граф без петель и кратных рёбер. Запрос в задаче представляет из себя пару (x, π), где $x \in \overline{\mathbb{R}}$, а π — обобщённый путь в графе G, где под обобщённым путём будем понимать последовательность вершин $v_1 v_2 \dots v_p$ и соответствующую ей последовательность рёбер $e_2 \dots e_p$, такие что для любого $2 \leqslant i \leqslant p$ ребро e_i соединяет вершину v_i с какой-либо из идущих до неё вершин v_j с $j < i$. Причём для любого ребра e_i, из обобщённого пути π выполнено $x \in R_{e_i}$. Задача состоит в том, чтобы для каждой вершины v_i пути π найти $\xi(x, C_{v_i})$ — первый элемент каталога C_{v_i}, значение которого не меньше, чем x.

Буквами d, n, m обозначим следующие характеристики графа каталогов L: $d = d(L)$ — локальная степень графа каталогов L, $n = |V|$ и $m = |E|$.

Для наглядности мы опишем три варианта решения этой задачи и не вполне формально отметим их плюсы и минусы. Первым вариантом будет последовательный бинарный поиск, вторым — общий суперкаталог и третьим — техника частичного каскадирования.

1) Последовательный бинарный поиск подразумевает следующую процедуру: мы просто проходим последовательно по пути π и для каждой встреченной вершины $v \in V$ бинарным поиском ищем $\xi(x, C_v)$. При этом не надо хранить никакой дополнительной информации, кроме самих каталогов.
2) Под общим суперкаталогом будем понимать каталог, в котором встречаются все элементы, встречающиеся во всех каталогах $\mathcal{C}$. Обозначим такой суперкаталог $\hat{C}$. Очевидно, что $\xi(x, \hat{C})$ однозначно определяет ответ и для других каталогов. Минусом такого подхода является то, что для каждого возможного результата поиска в суперкаталоге необходимо хранить набор ответов для всех каталогов C_v всех узлов $v \in V$.
3) Техника частичного каскадирования в некотором смысле является комбинацией первых двух вариантов. Идея здесь состоит в том, чтобы заменить граф каталогов $L = (G, \mathcal{C}, \mathcal{R})$, например,

на согласованную с L строгую систему мостов $S = (G, \mathcal{A}, \mathcal{R}, \mathcal{B})$. При этом каталог C_v каждого узла $v \in V$ фактически будет заменен на некоторый расширенный каталог A_v, который содержит $U(C_v)$ в качестве подкаталога. В дополнение для каждой вершины $v \in V$ необходимо в расширенном каталоге A_v для каждого возможного ответа хранить ссылку на соответствующий ответ C_v (так как $U(C_v)$ является подкаталогом A_v, то по значению $\xi(x, A_v)$ можно однозначно определить $\xi(x, C_v)$). Пусть при этом ширина системы мостов S равна g. Тогда обработку запроса (x, π) можно осуществлять следующим образом. Пусть $\pi = v_1 e_1 v_2 e_2 \dots v_{s-1} e_{s-1} v_s$. Сначала следует произвести бинарный поиск в расширенном каталоге A_{v_1} и по ссылке для ответа выдать $\xi(x, C_{v_1})$ — искомый ответ для каталога C_{v_1}. После этого для каждой последующей вершины пути v_i, в которую мы приходим из v_j по ребру e_i мы будем получать значение $\xi(x, C_{v_i})$ с помощью следующей процедуры.

(i) Начиная с элемента $\xi(x, A_{v_j})$ (значение $\xi(x, A_{v_j})$ определяется во время рассмотрения вершины v_j) перебираем все элементы каталога A_{v_j} по возрастанию, пока не встретим элемент, являющийся одним из концов перехода, принадлежащего мосту B_{e_i}, соединяющему каталоги A_{v_j} и A_{v_i}. Ниже показано, почему мы обязательно встретим такой элемент.

(ii) Переходим по найденному переходу к соответствующему элементу каталога A_{v_i}.

(iii) Идём по каталогу A_{v_i} в сторону убывания значений, пока не встретим в нём первый элемент, значение которого строго меньше, чем x (либо пока не дойдём до первого элемента A_{v_i}). Тем самым мы найдём первый элемент A_{v_i}, значение которого не меньше, чем x, то есть элемент $\xi(x, A_{v_i})$.

(iv) По соответствующей ссылке от $\xi(x, A_{v_i})$ определяем $\xi(x, C_{v_i})$.

Посмотрим, каково самое большое суммарное количество записей, которое нам потребуется перебрать во время этапов (3i) и (3iii). По условию $x \in R_{e_i}$, но отрезок покрытия моста B_{e_i} тоже равен R_{e_i} по определению системы мостов. Пусть $R_{e_i} = [a, b]$. Так как $a \leqslant x \leqslant b$, и в каталоге A_{v_j} есть элемент со значением b, лежащий на верхнем переходе моста B_{e_i}, то $\xi(x, A_{v_j}) \neq \Lambda$ и не лежит выше верхнего перехода моста B_{e_i}. Если $x > a$, то $\xi(x, A_{v_j})$ не может лежать и ниже нижнего перехода этого моста (и на самом нижнем переходе тоже). В этом случае $\xi(x, A_{v_j})$ либо принадлежит некоторому ярусу моста B_{e_i}, либо сразу лежит на некотором переходе этого моста (но не нижнем). В обоих случаях максимальное количество записей, которые мы можем посетить на этапах (3i) и (3iii) (не считая записей, соединяемым переходом, по которому мы идем на этапе (3ii)) не превосходит ширину моста, а она в свою очередь не больше g. Если же $x = a$, то $\xi(x, A_{v_j})$ либо лежит на нижнем переходе моста B_{e_i}, либо ниже него. Точно так же $\xi(x, A_{v_i})$ лежит либо на нижнем переходе того же моста (но с другой стороны), либо ниже него в каталоге A_{v_i}. Так как в каждом из каталогов A_{v_j} и A_{v_i} число записей со значением a не превосходит $1 + d$ и по одной такой записи лежит на нижнем переходе моста B_{e_i}, то фактически в этом случае на этапах (3i) и (3iii) мы посетим самое большее $2d$ записей без учета записей, лежащих на переходе, по которому мы идем на этапе (3ii). В обоих случаях число записей, которые мы просмотрим на этапах (3i) и (3iii) ограничено величиной $\max(g, 2d)$. Хранение каталогов A_v можно организовать так, что этапы (3ii) и (3iii) будут требовать константных затрат времени. В итоге время работы алгоритма оказывается пропорциональным времени поиска в расширенном каталоге A_{v_1} плюс на каждый последующий поиск требуется порядка $\max(g, 2d)$ операций. Увеличивая параметр g, можно получать меньший размер структур данных, однако время поиска в этих структурах будет возрастать. В оригинальной статье Чазелле и Гуибаса [1] рассматривается ограничение $g < g_0 = 6d - 1$, при котором объём требуемой памяти увеличивается по сравнению с последовательным бинарным поиском не более чем в 3 раза, а время перехода между соседними каталогами пропорционально d. Возможность варьирования g_0 в [1] упоминается, но не обсчитывается. В данной работе такое варьирование осуществляется.

Отметим, что существует модификация техники частичного каскадирования, с помощью которой время перехода между соседними каталогами можно уменьшить с пропорционального d до пропорционального $\log_2 d$. Для этого изначальный граф каталогов следует заменить на его модифицированный вариант, в котором каждая вершина меняется на целое поддерево таким образом, что модифицированный граф каталогов имеет локальную степень не более трёх. После чего обычный вариант техники частичного каскадирования применяется к модифицированному графу каталогов. Подробное описание того, как можно построить модифицированный граф каталогов приводится в работе [1].

4. Доказательство основного результата

Доказательство теоремы 1. Построим каталоги A_v и мосты B_e для искомой строгой системы мостов $S = (G, \mathcal{A}, \mathcal{R}, \mathcal{B})$ с помощью следующего алгоритма.

1) Для каждой вершины $v \in V$ положим соответствующий ей каталог A_v равным $U(C_v)$ — исходному каталогу C_v этой вершины, из которого удалили все дубликаты. Для каждого ребра $\{v, w\} = e \in E$ инициализируем соответствующий ему мост B_e между каталогами A_v и A_w следующим образом. Пусть ребру e соответствует отрезок $R_e = [a_e, b_e]$. Добавим в каталог A_v два новых элемента со значениями a_e и b_e соответственно. Под добавлением мы подразумеваем здесь добавление элемента в упорядоченный список с сохранением упорядоченности. При добавлении одного нового элемента индексы всех элементов каталога, идущих после добавленного вырастут на единицу, поэтому после добавления элемента в какой-либо из каталогов A_v мы будем соответствующим образом править содержимое каждого моста, связанного с A_v. Пусть добавленные в A_v элементы имеют индексы i_a и i_b соответственно. Аналогично добавим a_e и b_e в каталог A_w. Пусть добавленные туда элементы получили индексы j_a и j_b. Инициализируем мост $B_e = \{p_1, p_2\}$,

где $p_1 = (i_a, j_a)$, а $p_2 = (i_b, j_b)$. Очевидно, что p_1 и p_2 являются переходами и кроме того, отрезок покрытия моста B_e оказывается равен $[a_e, b_e] = R_e$. Нетрудно видеть, что, к моменту окончания этого шага, четвёрка $(G, \mathcal{A}, \mathcal{R}, \mathcal{B})$ будет являться строгой системой мостов (выполнение условия 3 из определения системы мостов докажем позднее), согласованной с начальным графом каталогов L. Условие (г) также выполнено, так как все элементы каталогов A_v, не принадлежащие ни одному переходу ни одного построенного моста это в точности все элементы каталога $U(C_v)$, которые мы добавили в самом начале.

2) Если все мосты B_e имеют ширину не превосходящую g_0, то остановимся. Если же хотя бы один мост B_e имеет ширину больше g_0, то у него есть некоторый ярус, ширина которого равна g и $g > g_0$. Проведём операцию разбиения этого яруса. Будем считать, что рассматриваемый мост B_e соединяет каталоги A_v и A_w. Выпишем все элементы рассматриваемого яруса обоих каталогов по возрастанию значений (но сохраняя порядок между одинаковыми элементами каждого каталога). Получится список J из g элементов. Положим $\sigma = (g_0 + 1)/2$ и разделим g на σ с остатком: $g = i\sigma + r$, где $0 \leqslant r < \sigma$. Но $g > g_0$, а значит $g \geqslant g_0 + 1 = 2\sigma$ и поэтому $i \geqslant 2$. Мы будем разбивать рассматриваемый ярус на $i - 1$ ярус ширины σ и один ярус ширины $\sigma + r$. Легко видеть, что $\sigma + r < 2\sigma = g_0 + 1$, а потому $\sigma + r \leqslant g_0$.

   Для того, чтобы произвести такое разбиение будем для каждого натурального числа t от 1 до $i - 1$ включительно производить следующую операцию. Обозначим как J_t список из $t\sigma$ первых элементов списка J. Пусть значение последнего элемента J_t равно $y_t \in \overline{\mathbb{R}}$. Вставим элемент со значением y_t в каталог A_v сразу после последнего элемента каталога A_v, попавшего в J_t (а если ни один элемент каталога A_v не попал в J_t, то новый элемент вставим сразу после конца нижнего перехода разбиваемого яруса). Упорядоченность каталога A_v при этом сохранится. Аналогично поступим с каталогом A_w. В итоге мы вставим по одному элементу со значением y_t в оба каталога A_v и A_w. Добавим переход, соединяющий эти два элемента к мосту B_e.

Такими действиями мы разобьём один ярус моста B_e с шириной $g > g_0$ на i ярусов, ширина которых не превосходит g_0. Заметим, что при этом в каждый из каталогов A_v и A_w добавится по $i - 1$ новому элементу, что может привести к увеличению ширин ярусов других мостов, соответствующих этим каталогам (соединяющих каталог A_v с какими-то другими каталогами, отличными от A_w или наоборот).

Очевидно, что $(G, \mathcal{A}, \mathcal{R}, \mathcal{B})$ все ещё строгая система мостов, согласованная с L (то, что условие 3 из определения системы мостов выполнено, будет доказано ниже). Условие (г) очевидно не будет нарушено в результате выполнения указанных выше операций.

Перейдём к началу шага 2.

Условие (в) выполняется на протяжении всех шагов алгоритма, поскольку нигде не используются какие-либо значения, кроме тех, что изначально содержались в каталогах из $\mathcal{C}$ и на концах отрезков из $\mathcal{R}$.

Легко видеть, что если алгоритм останавливается, то ширина каждого из построенных мостов не превосходит величины g_0, то есть выполнено условие (а). Осталось показать, что алгоритм всегда останавливается и что размеры построенных каталогов удовлетворяют условию (б).

Заметим, что каждый раз выполнение шага 2 приводит к добавлению как минимум двух новых записей в наши каталоги. Покажем, что в результате работы алгоритма общее количество записей в каталогах всегда удовлетворяет условию (б). Отсюда будет следовать, что шаг 2 не может выполняться бесконечное количество раз (иначе в какой-то момент ограничение на суммарное количество элементов перестало бы выполняться).

Чтобы показать, что условие (б) выполняется во время работы алгоритма, мы воспользуемся следующей техникой. Будем считать, что для добавления одного элемента в какой-нибудь из каталогов A_v нам требуется заплатить один *токен*. Токен будет у нас своеобразной валюты для покупки возможности добавить элемент. Валюту эту будем считать делимой на любые сколь угодно малые части, в частности $1/d$ часть токена будем называть *кредитом*. При добавлении каждого элемента в шаге инициализации мы будем одновременно выдавать ему $1+c$ токенов. Кроме того, в системе будут так называемые копилки, куда мы будем складывать временно не используемые токены, чтобы использовать их в дальнейшем. У каждого яруса каждого построенного моста будет своя копилка, для которой мы будем поддерживать следующее условие: если ширина яруса равна g, то в его копилке должно находиться не меньше $c\max(0, g-\sigma)$ кредитов. Рассмотрим подробнее первый этап нашего алгоритма. Во время его выполнения в каталоги добавляется $\sum_{v\in V}|U(C_v)| + 4m$ элементов (второе слагаемое получается таким из-за того, что для каждого ребра $e \in E$, которому соответствует отрезок $R_e = [a_e, b_e]$ мы должны добавить в два каталога по одному элементу со значением a_e и по одному элементу со значением b_e). Как мы уже говорили, каждому элементу, добавленному на шаге инициализации, мы выдаём $1+c$ токен. 1 токен сразу тратится на то, чтобы добавить сам элемент. Остаётся по c токенов на элемент. После инициализации каждый элемент может лежать на нескольких разных ярусах (обязательно разных мостов, так как принадлежать сразу двум ярусам одного моста элемент не может). Покажем, что элемент не может принадлежать более чем d ярусам. Предположим, что это не так и некоторый элемент каталога A_v со значением $x \in \overline{\mathbb{R}}$ принадлежит каким-то ярусам различных мостов $B_{e_1}, \ldots, B_{e_l}$, где $l > d$. Но тогда получалось бы, что для любого i имеет место $x \in R_{e_i}$. А это противоречит определению числа d. Таким образом, каждый элемент любого из наших каталогов принадлежит не более чем d различным ярусам. Вспомним, что на шаге инициализации у каждого элемента осталось неиспользованными c токенов, что равно dc кредитам. В копилку каждого из ярусов, в которые попал рассматриваемый элемент положим c кредитов (оставшихся кредитов должно хватить). Если какая-то часть кредитов осталась неиспользованной, просто выкинем их. В итоге после выполнения шага инициализации окажется, что в копилке каждого яруса ширины g лежит в точности gc кредитов (каждый добавленный в ярус элемент приносил с собой по c кредитов), что не меньше требуемой величины $c \cdot \max(0, g-\sigma)$. Покажем, что это условие можно продолжить соблюдать и в дальнейшем при многократном выполнении шага 2. Для этого посмотрим на процедуру разбиения яруса ширины g с точки зрения нашей валюты. Итак, в начале шага 2, по нашему предположению, выполняется условие, гласящее, что в копилке рассматриваемого яруса находится не меньше $c(g-\sigma)$ кредитов. Эти кредиты мы будем сейчас использовать: часть из них пойдёт на добавление новых элементов в каталоги, другая часть пойдёт в копилки новых ярусов, заменяющих один разбиваемый. В обозначениях шага 2 имеет место $g = i\sigma + r$. Поэтому в нашей копилке имеется как минимум $c(i\sigma + r - \sigma) = c(i-1)\sigma + cr$ кредитов. Вспомним, как делится ярус. Фактически один рассматриваемый ярус разделяется на $i-1$ ярус ширины σ и один ярус ширины $\sigma + r$. В копилку ярусов ширины σ мы не будем класть ни одного кредита, а в ярус ширины $\sigma + r$ положим cr кредитов. От нашей копилки останется не меньше $c(i-1)\sigma$ кредитов. Вспомним также, что во время разбиения нам пришлось добавить по $i-1$ элементу сразу в два каталога, то есть в сумме $2(i-1)$ элементов. На каждый такой элемент нам достаточно потратить $1+c$ токен: 1 токен пойдёт непосредственно на добавление элемента в соответствующий каталог, а оставшиеся c токенов или что то же самое cd кредитов пойдут на то, чтобы добавить по c кредитов в копилку всех ярусов, в которые попал новый элемент (их не более d — доказательство полностью аналогично приведённому выше доказательству, что после инициализации каждый элемент принадлежит не более чем d ярусам). То есть нам потребуется $2(i-1)(1+c)$ токен или $2(i-1)(1+c)d$ кредитов. Но $c(i-1)\sigma = c(i-1)(g_0+1)/2 = (i-1)\frac{4d}{g_0+1-4d}\frac{g_0+1}{2} = (i-1)2d\frac{g_0+1}{g_0+1-4d} = 2(i-1)d(1+\frac{4d}{g_0+1-4d}) = 2(i-1)(1+c)d$. Отсюда следует, что кредитов из копилки разбиваемого яруса хватит и на добавление новых элементов (вспомним, что часть нашей валюты в количестве $2(i-1)$ токенов ушла из системы копилок, когда мы добавляли такое же количество новых элементов), и на поддержание ограничения на размеры копилок.

Итак, что мы получили: токены поступали в систему извне только на первом шаге в количестве $1+c$ токен на каждый добавленный элемент, а всего таких элементов было $\sum_{v\in V}|U(C_v)| + 4m$. Во время последующих проходов по шагу 2 новые токены не поступают — мы лишь перекладываем токены из одной копилки в другую, плюс тратим не менее 2 токенов. Из этого следует, что шаг 2 выполняется лишь конечное число раз. Кроме того, в систему копилок поступило $(1+c)(\sum_{v\in V}|U(C_v)|+4m)$ токенов, а так как на добавление в каталог одного элемента затрачивается один токен, то имеет место $\sum_{v\in V}|A_v| \leqslant (1+c)(\sum_{v\in V}|U(C_v)|+4m)$.

Осталось показать, что после каждого шага нашего алгоритма выполняется условие 3 из определения системы мостов, то есть тот факт, что для любого $x \in \overline{\mathbb{R}}$ ни один из мостов B_e не содержит двух x-переходов. Далее мост между каталогами будем называть x-мостом, если он содержит хотя бы один x-переход. Для доказательства условия 3 покажем, что после шага инициализации, а также после каждого выполнения шага 2 выполнены следующие утверждения:

1) Для любого $x \in \overline{\mathbb{R}}$ у каждого моста из $\mathcal{B}$ есть не более одного x-перехода;
2) Ни один из каталогов A_v не содержит более чем $1+h$ записей с каким-то одним значением x, где h — количество x-мостов, ведущих из или в v (из определения d следует, что $h \leqslant d$).

Сначала покажем, что первое утверждение выполнено после шага инициализации. Это следует из того, что после инициализации любой мост состоит ровно из двух переходов, причём если мост соответствует ребру графа e с отрезком $R_e = (a_e, b_e)$, то значения элементов на переходах этого моста равны a_e и b_e. По условию $a_e < b_e$, поэтому $a_e \neq b_e$ и ни одно значение не встречается на переходах одного моста дважды.

Покажем, что в любой момент после шага инициализации и после очередного выполнения шага 2 второе утверждение следует из первого. Действительно, в каталог A_v элементы со значением x могут быть добавлены только по следующим причинам: во-первых, один такой элемент добавляется во время инициализации если элемент со значением x присутствует в каталоге $U(C_v)$, и во-вторых, по одному такому элементу добавляется для каждого x-перехода каждого моста ведущего из или в v. Из первого утверждения следует что для каждого моста, ведущего из или в v таких переходов может быть не более одного. Таким образом, каталог A_v может содержать самое большее $1+h$ элементов со значением x, где h — количество x-мостов, ведущих из или в v.

Покажем, что если перед началом шага 2 оба утверждения выполнены, то после него выполнено первое утверждение. Предположим, что это не так и после выполнения очередной итерации шага 2 первое утверждение перестало выполняться. Это означает, что в результате разбиения некоторого яруса некоторого моста рассматриваемый мост стал содержать более одного x-перехода для некоторого $x \in \overline{\mathbb{R}}$. Рассмотрим два возможных случая.

1) До разбиения мост уже содержал x-переходы. Так как по условию до разбиения выполнялось первое утверждение, то мост мог содержать только один x-переход. Очевидно, что если происходит разбиение яруса, нижний и верхний переходы которого являются a- и b-переходами соответственно, то значения соединяемых новыми переходами элементов должны лежать в интервале от a до b. Так как после разбиения мост содержит более одного x-перехода, то единственный до разбиения x-переход рассматриваемого моста является либо верхним, либо нижним переходом разбиваемого яруса. Рассмотрим оба этих случая, заметив что в каждом из них разбиваемый ярус содержит самое большее $2d$ элементов со значением x (суммарно в обоих каталогах, соединяемых рассматриваемым мостом содержится не более $2(1+d)$ таких элементов, но два элемента не принадлежат ни одному из ярусов моста, так как составляют x-переход).

   a) Единственный x-переход является нижним переходом разбиваемого яруса. Рассмотрим самое маленькое значение, которое могут иметь элементы, соединяемые переходами, добавленными во время разбиения. Вспомним, как определялось это значение. Мы брали список J, составленный из всех элементов разбиваемого яруса. Значение новых элементов, которые добавлялись в оба каталога и соединялись переходом, было равно значению элемента, оказавшемуся в этом списке под номером $\sigma = (g_0+1)/2 \geqslant (4d+2)/2 = 2d+1$. Так как все элементы яруса имеют значение не меньше x, а элементов со значением x в ярусе не более $2d$, то это будет значение строго большее x. Таким образом, x-переход в этом случае не может быть добавлен, поэтому данный случай не может иметь места.

б) Единственный x-переход является верхним переходом разбиваемого яруса. Аналогичным образом рассмотрим самое большое значение, которое могут иметь элементы, соединяемые переходами, добавленными во время разбиения. Это значение определяется значением последнего элемента того же списка J, из которого выкинули не менее $\sigma \geqslant 2d+1$ самых больших элементов. Так как все элементы яруса не превосходят x и элементов со значением x в ярусе не более $2d$, то также получаем, что x-переход добавлен не будет и этот случай также невыполним.

2) До разбиения мост не содержал ни одного x-перехода. Так как после разбиения у моста появились x-переходы и даже более одного, то x должен был принадлежать интервалу покрытия рассматриваемого моста. Но рассматриваемый мост не был x-мостом, а значит у обеих вершин v и w, соединяемых рассматриваемым мостом, количество x-мостов было не более $d-1$. По первому утверждению отсюда следует, что в каждом из каталогов A_v и A_w было не более d элементов со значением x. А отсюда следует, что в разбиваемом ярусе могло быть самое большее $2d$ таких элементов. Используя неравенство на минимальную ширину итогового яруса $\sigma \geqslant 2d+1$ аналогично только что рассматриваемым вариантам получим, что к мосту мог быть добавлен только один x-переход и после выполнения разбиения в нём не может оказаться более одного x-перехода.

Во всех рассмотренных случаях мы пришли к противоречию. Значит, наше предположение было неверно и после выполнения шага 2 первое утверждение остаётся справедливым. По индукции получаем, что оба утверждения остаются справедливыми после каждой итерации. Таким образом, четвёрка $(G, \mathcal{A}, \mathcal{R}, \mathcal{B})$ на протяжении работы алгоритма остаётся строгой системой мостов, согласованной с графом каталогов L. Теорема доказана.

Доказательство следствия 1. Применяем сначала теорему 1 и с её помощью получаем строгую систему мостов $S' = (G, \mathcal{A}', \mathcal{R}, \mathcal{B}')$, согласованную с графом каталогов L. Строим искомую систему мостов $S = (G, \mathcal{A}', \mathcal{R}, \mathcal{B}')$ следующим образом: каждый каталог $A'_v \in \mathcal{A}'$ заменяем на $A_v = U(A'_v)$, а мост $B'_e \in \mathcal{B}'$ (где $e = \{v, w\}$) заменяем на мост B_e, состоящий из такого же количества переходов, причём каждый x-переход моста B'_e заменяется на единственно возможный x-переход, соединяющий единственный элемент со значением x в каталоге A_v с единственным элементом со значением x из каталога A_w. Так как S' являлась системой мостов, то каждый мост в ней содержал не более одного x-перехода, а потому построенный таким образом B_e будет являться мостом между A_v и A_w. Отрезок покрытия моста B_e очевидно совпадает с отрезком покрытия исходного моста B'_e. Таким образом, легко видеть, что S является системой мостов (хотя уже и не обязательно строгой). Из того, что система мостов S' была согласована с графом каталогов L также очевидно, что S тоже согласована с L. Требуемые условия на S следуют из построения и из свойств системы S'.

Если же теперь допустить, что все отрезки $R_e \in \mathcal{R}$ равны между собой и равны некоторому отрезку $[a, b]$, то каталоги A'_v должны были содержать по меньшей мере $2m$ элементов со значением a и не меньше $2m$ элементов со значением b. В системе мостов S каждый каталог A_v очевидно содержит не более одного элемента со значением a и не более одного со значением b. Итого, вместо не менее $4m$ элементов мы получаем самое большее $2n$ элементов. Таким образом, получаем $\sum_{v \in V} |A_v| \leqslant (1 + c)(\sum_{v \in V} |U(C_v)| + 4m) - 4m + 2n = (1 + c) \sum_{v \in V} |U(C_v)| + 2n + 4mc$.

Список литературы

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

[2] Chazelle B., Guibas L. J. Fractional cascading: II. Applications // Algorithmica. 1. 1986. P. 163–191.