

О ВЕРХНЕЙ ГРАНИЦЕ СЛОЖНОСТИ СОРТИРОВКИ¹⁾

© 2021 г. И. С. Сергеев

125438 Москва, 4-й Лихачёвский пер., 15, ФГУП “НИИ “Квант”, Россия

e-mail: isserg@gmail.com

Поступила в редакцию 18.09.2019 г.
Переработанный вариант 23.07.2020 г.
Принята к публикации 16.09.2020 г.

Показано, что для сортировки набора из n элементов линейно упорядоченного множества всегда достаточно $\log_2(n!) + o(n)$ попарных сравнений. Библ. 14. Фиг. 3.

Ключевые слова: сортировка, сложность, дерево решений, частичный порядок, симплекс.

DOI: 10.31857/S0044466921020125

1. ВВЕДЕНИЕ

Рассматривается стандартная задача сортировки набора из n элементов линейно упорядоченного множества с помощью попарных сравнений. Подробное введение в проблематику представлено в [1, гл. 3], [2, п. 5.3], [3, Ch. 2].

Алгоритм сортировки можно изобразить в виде бинарного корневого дерева с ориентацией ребер в направлении от корня. Такое дерево обычно называется деревом сравнений, а также деревом решений или бинарной решающей диаграммой. Внутренняя вершина дерева соответствует операции сравнения некоторых двух элементов. В зависимости от результата сравнения алгоритм осуществляет переход по одному из ребер к следующей вершине. Концевая вершина (лист) дерева соответствует полученному в результате сравнений упорядочению входного набора. *Сложностью* алгоритма называется глубина дерева – максимальное расстояние в ребрах между корнем и листом.

Естественно ограничить рассмотрение алгоритмами, не выполняющими избыточных сравнений (т.е. сравнений, не добавляющих новой информации о порядке элементов). В соответствующих таким алгоритмам деревьях каждая возможная перестановка связана ровно с одним листом. В частности, деревья для сортировки n -элементного набора имеют ровно $n!$ листьев.

Пусть $S(n)$ означает минимальную сложность алгоритма сортировки n -элементного набора. Поскольку глубина дерева с m листьями не меньше $\log m$, имеет место простая нижняя оценка

$$S(n) \geq \log(n!) = n \log n - \log e \cdot n + O(\log n), \quad (1)$$

$\log e \approx 1.443$. (Здесь и далее по тексту основание у двоичных логарифмов опускается.) Более того, поскольку даже средняя глубина дерева по всем его листьям не превосходит $\log m$, то оценка (1) действительна и для сложности сортировки в среднем по всем $n!$ возможным перестановкам входного набора (см. также [1]–[3]). Подобные нижние оценки называются теоретико-информационными.

С точки зрения верхней оценки, в целом лучшим среди известных остается метод Форда–Джонсона [4] (метод бинарных вставок), предложенный более 60 лет назад. Он приводит к соотношению

$$S(n) \leq \log(n!) + cn + O(\log n), \quad (2)$$

где константа c в зависимости от n варьируется от $\log(3e/8) \approx 0.028$ (в благоприятном случае $n \sim 2^k/3$) до $\log[3/(4 \ln 2)] \approx 0.114$ (в неблагоприятном случае $n \sim \ln 2 \times 2^k/3$). Метод бинарных вставок доказуемо оптимален при $n \leq 15$ и при некоторых больших n (см. [5]). Усовершенствование метода, предложенное в [6] около 1980 г., позволяет уточнить константу в неблагоприят-

¹⁾Работа выполнена при финансовой поддержке РФФИ (код проекта 19-01-00294а).

ном случае до $c \approx 0.105$. Достаточно большую работу по сглаживанию неравномерности оценок метода бинарных вставок проделали авторы [7]. По результатам этой работы можно указать константу для неблагоприятного случая где-то в пределах $0.06 < c < 0.07$, точнее сказать трудно¹. Для сложности сортировки в среднем оценки сближены гораздо сильнее. Константа в верхней границе сложности недавно была понижена до $c \approx 0.032$ при любом n (см. [8], [9]).

Метод Форда–Джонсона основан на процедурах вставки элементов в линейно упорядоченное множество. Вставка каждого элемента выполняется отдельно. Эффективность метода обеспечивается путем подбора мощности целевого множества, близкой к степени двойки.

Однако известно, что совместная вставка даже двух элементов выполняется быстрее, чем раздельная, если мощность n целевого множества находится в пределах $2^k < n < \frac{17}{14} \times 2^k - 1$ (см. [10], [11], а также [2]). Еще выгоднее может быть группировка элементов по 4 или 5 с последующей их сортировкой до вставки (см. [6]). Высокая эффективность метода работы [8] (с точки зрения средней сложности) также достигается за счет того, что часть вставок выполняется совместно для пар элементов.

В развитие этой идеи мы предлагаем алгоритм групповой вставки большого числа элементов, организованный в некотором смысле как система массового обслуживания. Этот подход переключается с концепцией массового производства (частично упорядоченных наборов), которой следуют лучшие известные алгоритмы выбора элемента заданного порядка (см. [12], [13]). Предлагаемый метод позволяет приблизить сложность вставки, приходящуюся на один элемент, к теоретико-информационной нижней границе, т.е. $\log n + o(1)$ при любом n . Как следствие, на методе сортировки с помощью групповых вставок достигается оценка

$$S(n) \leq \log(n!) + o(n).$$

Разумеется, такая же оценка справедлива и для сложности сортировки в среднем.

Изложение построено следующим образом. В разд. 2 дается краткая справка о методе Форда–Джонсона. В разд. 3 приводятся некоторые элементарные соображения, лежащие в основе предлагаемого метода. Обобщенная концепция метода групповой вставки изложена в разд. 4. Центральная часть метода – стратегия выбора элементов для сравнений – описана в разд. 5. В разд. 6 представлены основные заключения о сложности групповой вставки и сортировки.

2. МЕТОД БИНАРНЫХ ВСТАВОК

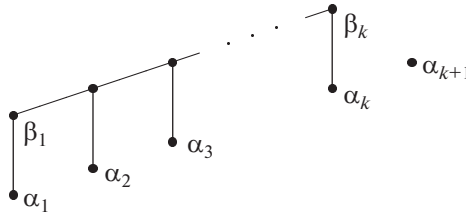
Операцию сравнения элементов e_1 и e_2 будем обозначать через $e_1 ? e_2$, а отношение порядка – неравенствами $e_1 < e_2$ или $e_1 > e_2$. Если элементы e_1 и e_2 совпадают, то операция сравнения все равно возвращает либо $e_1 < e_2$, либо $e_1 > e_2$. Без ограничения общности можно считать, что все элементы входного набора различны. Для краткости линейно упорядоченный набор элементов далее будем называть *цепочкой*. Под *длиной* цепочки будем понимать число элементов в ней. Номер элемента цепочки при нумерации с 1 в порядке возрастания будем называть *рангом*.

Напомним суть метода бинарных вставок (см. [4], а также [2]): все элементы разбиваются на пары, внутри пар выполняются сравнения, большие элементы пар сортируются. В результате получается частичный порядок, диаграмма² которого изображена на фиг. 1: элементы в парах обозначены через α_i, β_i , где $\beta_i > \alpha_i$, нумерация в порядке возрастания β_i . В нечетном случае $n = 2k + 1$ один из элементов не имеет пары, он обозначен через α_{k+1} .

По построению элементы $\alpha_1, \beta_1, \beta_2, \dots, \beta_k$ образуют цепочку. Далее проводится последовательная вставка оставшихся элементов α_i в эту цепочку. Первым вставляется элемент α_3 с помощью двух сравнений, затем элемент α_2 – также за два сравнения; далее вставляются все элементы, для которых достаточно трех сравнений (это α_5 и α_4 в таком порядке) и т.д. Всякий раз вставка элемента выполняется в цепочку длины $2^j - 1$, где $j = 2, 3, \dots$, разве что в финальной серии вставок используется цепочка неполной длины.

¹ Наиболее сильная форма результата в [7] представлена графически.

² Диаграмма Хассе: ребра соединяют пары элементов с известным порядком (подробнее см., например, [2]).



Фиг. 1. Частичный порядок в методе бинарных вставок.

Можно проверить, что j сравнений достаточно для вставки каждого из элементов α_i с номерами $u_{j-1} < i \leq u_j$, где $u_j = \frac{2^{j+1} + (-1)^j}{3}$. Таким образом, сложность $f(n)$ сортировки n -элементного набора методом Форда–Джонсона при $\lfloor 2^{k+1}/3 \rfloor \leq n < \lfloor 2^{k+2}/3 \rfloor$ описывается выражением

$$f(n) = kn - \lfloor 2^{k+2}/3 \rfloor + \lfloor k/2 \rfloor + 1.$$

Эту формулу удобно переписать в асимптотическом виде. Пусть $n = \tau \times 2^{k+1}/3$, где $\tau \in [1, 2)$. Тогда

$$f(n) = n \log n - [2/\tau + \log(2\tau/3)]n + \log n/2 + O(1). \tag{3}$$

Коэффициент в скобках при линейном члене принимает максимальное значение $\log(8/3) \approx 1.415$ при $\tau = 1$, а минимальное значение $\log(4e \ln 2/3) \approx 1.329$ — при $\tau = 2 \ln 2$. Более подробный анализ метода, см. в [4], [2].

Введем стандартное обозначение $M(m, n)$ для сложности слияния двух цепочек длины m и n . Одним из результатов работы Шульте–Мёнтинга [6] является оценка $M(5, n) \leq 5k - 8$ при $n \leq \frac{319}{448} \times 2^k - O(1)$. Она демонстрирует возможность вставки пятерки элементов в цепочку длины n за $5k - 8 + S(5) = 5k - 1$ сравнений, т.е. со средней сложностью $k - 1/5$ на один элемент. Поэтому в общем случае часть вставок финальной серии метода Форда–Джонсона, а именно, вставку элементов с номерами после u_{k-1} , выгодно произвести в цепочку меньшей длины со средней сложностью $k - 1/5$ вместо k . Это наблюдение приводит к уточнению оценки (3) для всех n , за исключением очень близко расположенных к точкам последовательности $\{2^i/3\}$. Максимальное значение константы в оценке сложности (2) при этом понижается до $c \approx 0.105$. Аналогичный результат получен в [7] весьма сложным способом: вместо вставок в финальной стадии алгоритма используются слияния длинных цепочек методом Кристена (см. [14]), но при этом оценка для неблагоприятного случая улучшена значительно.

3. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Для удобства изложения введем еще несколько понятий. В задачах слияния или вставки более длинную цепочку будем называть *главной*. Множество элементов главной цепочки, расположенных между некоторыми элементами α и β , назовем *интервалом* и обозначим через (α, β) : если Z — главная цепочка, то $(\alpha, \beta) = \{z \in Z \mid \alpha < z < \beta\}$. В качестве концов интервала также допускаются условные элементы $\pm\infty$: по определению, $-\infty < \alpha < +\infty$ выполнено для всех α . В отличие от длины цепочки, *длину интервала* определим как увеличенное на единицу число элементов в нем, т.е. как число возможностей для вставки нового элемента. Для длины будем использовать обозначение $|(\alpha, \beta)|$.

Обозначим через $Q(a)$ максимальное число n , при котором вставка элемента в интервал длины n выполняется со сложностью не более a или, иначе говоря, $M(1, n - 1) \leq a$. Тривиально,

$$Q(a) = 2^{\lfloor \log a \rfloor}. \tag{4}$$

Аналогично, через $P(a)$ обозначим максимальное n такое, что $M(2, n-1) \leq 2a$: в аргументе функции P записываем число сравнений, приходящееся на один элемент вставляемой пары. Известно (см. [10], [11]), что

$$P(a) = \begin{cases} \left\lfloor \frac{17}{14} \times 2^a \right\rfloor, & a \in \mathbb{N}, \\ \left\lfloor \frac{12}{7} \times 2^{a-1/2} \right\rfloor, & (a-1/2) \in \mathbb{N}. \end{cases} \quad (5)$$

Пусть $M(m \times k, n)$ означает сложность слияния m цепочек длины k с цепочкой длины n . Введем обозначения $Q_m(a)$ и $P_m(a)$ для максимального числа n , при котором $M(m \times 1, n-1) \leq am$ и соответственно $M(m \times 2, n-1) \leq 2am$.

По определению, $Q_1(a) = Q(a)$ и $P_1(a) = P(a)$. Также очевидно, что $Q_m(a) \geq Q(a) - m + 1$ и $P_m(a) \geq P(a) - 2(m-1)$.

Первое следствие, оправдывающее введение функции Q_m , можно извлечь из (5). Поскольку $Q_2(a) \geq P(a-1/2)$,

$$Q_2(a) \geq \begin{cases} 2^a - 1, & a \in \mathbb{N}, \\ \left\lfloor \frac{17}{14} \times 2^{a-1/2} \right\rfloor, & (a-1/2) \in \mathbb{N}. \end{cases} \quad (6)$$

Таким образом, $Q_2(a)$ может быть существенно больше, чем $Q(a)$.

При $m=1$ осмысленны только полуцелые аргументы у функций $P_m(a)$ и $Q_{2m}(a)$. Скажем, $P_1(r-1/4) = P_1(r-1/2) \approx \frac{6}{7} \times 2^r$, $r \in \mathbb{N}$. Однако с ростом m открываются новые возможности. Для разминки и иллюстрации основной идеи предлагаемого далее метода выведем оценку

$$P_m\left(r - \frac{1}{4} + \frac{1}{4m}\right) \geq \frac{51}{56} \times 2^r - 4m. \quad (7)$$

В главной цепочке длины $\left\lfloor \frac{51}{56} \times 2^r \right\rfloor - 4m$ выберем элемент α ранга $\left\lfloor \frac{17}{56} \times 2^r \right\rfloor - 2m + 1$. Тогда интервал $(-\infty, \alpha)$ имеет длину не более $Q_2(r-3/2) - 2(m-1)$, согласно (6), а интервал $(\alpha, +\infty)$ – не более $P(r-1) - 2(m-1)$, согласно (5).

Произвольная пара $\beta_0 < \beta_1$ обрабатывается следующим образом. Сравниваем β_0 с α . Если $\beta_0 > \alpha$, то пара вставляется в интервал $(\alpha, +\infty)$ за $2(r-1)$ сравнений методом из [10], [11]. Иначе, пара разбивается: элемент β_1 вставляется в главную цепочку бинарным методом за r сравнений, а элемент β_0 помещается во временное хранилище – *контейнер*. Если контейнер был пуст, то переходим к обработке следующей пары. Но если в контейнере оказалось 2 элемента, вставляем их в интервал $(-\infty, \alpha)$ за $2r-3$ сравнений.

По мере выполнения вставок главная цепочка и ее подынтервалы удлиняются. Выбор длин интервалов с запасом $2(m-1)$ обеспечивает возможность вставки в них элементов всех m обрабатываемых пар за планируемое число сравнений: соответственно $2(r-3/2)$ для двух элементов в интервал $(-\infty, \alpha)$ и $2(r-1)$ – для пар в интервал $(\alpha, +\infty)$.

После того, как обработка всех пар завершена, в контейнере еще может оставаться один элемент. Тривиальным способом он вставляется в цепочку за $r-1$ сравнений. Тогда общее число сравнений, выполняемых алгоритмом, в худшем случае оценивается как $m + m(r + r - 3/2) + 1/2 = 2m(r - 1/4) + 1/2$. Соотношение (7) доказано.

4. ОБЩИЙ МЕТОД

В этом разделе мы опишем общий вид предлагаемого метода групповой вставки вместе со средствами его анализа, попутно вводя необходимые понятия.

Операцию сравнения в любом методе вставки или слияния можно рассматривать как шаг разбиения главной цепочки на все меньшие интервалы, в которых локализуются вставляемые элементы или группы элементов. Это дает возможность рекурсивного построения алгоритма: задача слияния с длинной цепочкой с помощью нескольких сравнений сводится к слияниям с более короткими цепочками.

Опишем основные принципы метода групповой вставки упорядоченных пар элементов в главную цепочку. В главной цепочке выделяются интервалы, которые называются *финишными*.

Вставка элемента или пары элементов в финишный интервал выполняется подходящим алгоритмом из доставляющих оценки (4)–(6). Для некоторых финишных интервалов предусмотрены контейнеры. Как в простом примере выше, любой контейнер имеет емкость 2 элемента. Когда в контейнере оказывается два элемента, то выполняется их совместная вставка в соответствующий интервал.

Пары обрабатываются по очереди. Для результата обработки пары могут быть следующие возможности: пара попадает в финишный интервал и применяется метод из [10], [11] для окончательной вставки ее в главную цепочку, либо пара в какой-то момент разбивается и элементы обрабатываются по отдельности.

Судьба одиночных элементов может быть следующей: либо элемент попадает в финишный интервал и вставляется в главную цепочку бинарным методом, либо элемент помещается в контейнер. Если в контейнере нет других элементов, то работа продолжается с новой парой. Иначе выполняется обработка двух элементов из контейнера.

После того, как все пары подверглись обработке, любой из вставляемых элементов либо включен в главную цепочку, либо находится один в некотором контейнере. Алгоритм завершается тем, что каждый из оставшихся в контейнерах элементов вставляется в свой интервал тривиальным бинарным методом.

Удельной сложностью обработки одной пары назовем число сравнений, приходящихся на элементы пары, в наихудшем случае при условии, что ни один из элементов пары не остался в контейнере (для этого можно предположить, что все контейнеры не пусты). В ходе подсчета сложности одно сравнение, выполняемое при совместной обработке пары элементов, засчитывается как полсравнения каждому элементу. Удельная сложность вставки пары в примере из предыдущего раздела равна $2(k - 1/4)$.

Итоговую сложность алгоритма групповой вставки можно оценить сверху как

$$\rho m + C \log(n + 2m), \quad (8)$$

где ρ – удельная сложность, m – число вставляемых объектов, n – длина исходной цепочки, а C – общее число контейнеров. Избыточное число сравнений при вставке последнего элемента в контейнере здесь оценивается грубо как $\log(n + 2m)$; более аккуратное рассуждение привело бы к оценке $O(1)$; в реальных алгоритмах эта величина не превосходит 1, как в примере из разд. 3.

Теперь введем комплексное понятие стратегии сравнений. Пусть задан некоторый числовой набор $y_1, \dots, y_d \in \mathbb{R}$ с условием $0 \leq y_1 < \dots < y_d < 1$. Вводятся величины $p_{r,i}$, характеризующие длину интервала $J_{r,i}$, вставка пары в который возможна с удельной сложностью $2(r + y_i)$. Наряду с величинами $p_{r,j}$ будем также использовать удельные величины $x_{r,j} = p_{r,j} \times 2^{-r}$.

При любом $i = 1, 2, \dots, d$ и $r \geq r_0$ задается сокращенное дерево сравнений $T_{r,i}$ для вставки пары в интервал $J_{r,i}$. Под сокращенным деревом мы понимаем поддерево обычного дерева сравнений. Концевые вершины $T_{r,i}$ соответствуют вызову процедуры вставки в интервалы меньшей длины, назовем эти интервалы *терминальными*. Разбиение пары допускается только в концевых вершинах дерева. Так определенные сокращенные деревья сравнений описывают сведение задачи вставки в длинный интервал к задачам меньшей размерности.

Концевой вершине дополнительно ставятся в соответствие ограничения на длины терминальных интервалов, обеспечивающие выполнение оценки удельной сложности вставки пары в интервал $J_{r,i}$. Если при концевой вершине на расстоянии s от корня выполняется вставка пары в терминальный интервал (α', α'') , то ограничение имеет вид $|(\alpha', \alpha'')| \leq p_{l,j}$, где $l + y_j + s/2 \leq r + y_i$. Для одиночного элемента β , вставляемого в (α', α'') , ограничение может выглядеть либо как $|(\alpha', \alpha'')| \leq p_{l,j}$, либо как $|(\alpha', \alpha'')| \leq 2^l$. В первом случае вклад этого элемента в удельную сложность оценивается как $l + y_j + s/2 + 1/2$, а во втором – как $l + s/2$. Сумма оценок для двух элементов пары не должна превосходить $2(r + y_i)$.

Будем считать, что ранги элементов α', α'' также выражаются линейными комбинациями величин $p_{l,j}$. Некоторые ограничения могут при этом тривиально выполняться (например, в случае тождественности правой и левой частей неравенств). Множество оставшихся определим как *систему ограничений* стратегии.

Итак, под *стратегией сравнений* будем понимать семейство деревьев (фактически алгоритмов) $T_{r,i}$ и систему ограничений, обеспечивающую удельную сложность $2(r + y_i)$ вставки пар в интервалы $J_{r,i}$. *Глубиной* стратегии назовем максимальную глубину деревьев $T_{r,i}$.

Если минимальная удельная сложность вставки пары по всем терминальным интервалам в алгоритме с деревом сравнений $T_{r,i}$ равна $2(r' + y_j)$, то положим $w_{r,i} = r - r' + 1$.

Мы ограничим рассмотрение *однородными* стратегиями – такими, в которых очередной элемент интервала $J_{r,i}$, выбираемый для сравнения, делит интервал $J_{r,i}$ в отношении, не зависящем от r (с точностью до целочисленного округления). Тогда при любом i все деревья $T_{r,i}$ подобны некоторому дереву T_i . Можно считать, что внутренней вершине T_i приписана пара (y, Δ) , $y \in \{0, 1\}$, $\Delta \in \mathbb{R}$, кодирующая сравнение $\beta_y ? \alpha$, где (β_0, β_1) – вставляемая пара, а α – элемент ранга $\Delta |J_{r,i}|$ в интервале $J_{r,i}$. В однородной стратегии $w_{r,i} = w_i$ при всех r . Величину $\max_i w_i$ назовем *шириной* однородной стратегии. Ширина стратегии характеризует максимальную разницу сложности вставки между исходным интервалом и его терминальными подынтервалами. Ниже под стратегией всюду понимается однородная стратегия.

Далее в выкладках позволим длинам интервалов и рангам элементов цепочки принимать нецелые значения, под которыми будут пониматься округления до ближайшего целого в ту или иную сторону. В случае утверждения о сложности вставки в интервал нецелой длины x будем требовать выполнение этого утверждения для интервала длины $\lceil x \rceil$.

Наша ближайшая цель – построить переход от стратегии к конкретному алгоритму. В предположении существования пределов $z_i = \lim_{r \rightarrow \infty} x_{r,i}$ ограничения стратегии можно переписать в терминах предельных величин z_i и перейти к задаче линейного программирования. Ниже следуют описание и обоснование корректности такого перехода.

Пусть на множестве последовательностей d -мерных вещественных векторов $x_0, x_1, \dots \in \mathbb{R}^d$, $x_j = (x_{j,1}, x_{j,2}, \dots, x_{j,d})$, определена система σ нормированных линейных неравенств ширины w вида

$$\sum_{i=1}^d \sum_{j=0}^{w-1} a_{i,j,k} x_{t+j,i} \leq b_k, \quad \text{где} \quad \sum_{i=1}^d \sum_{j=0}^{w-1} |a_{i,j,k}| = 1, \quad k = 1, 2, \dots, K, \tag{9}$$

при всех $t \geq 0$. Здесь $a_{i,j,k}$, b_k – вещественные коэффициенты. С помощью формальной подстановки $x_{j,i} = z_i$ для всех i, j в (9) получим *приведенную систему*

$$\sum_{i=1}^d a_{i,k} z_i \leq b_k, \quad a_{i,k} = \sum_{j=0}^{w-1} a_{i,j,k}, \quad k = 1, 2, \dots, K, \tag{10}$$

относительно переменных z_i .

Система (10) определяет симплекс $T(\sigma)$ в пространстве \mathbb{R}^d . Для произвольного параметра $\varepsilon \geq 0$ определим вложенный симплекс $T^\varepsilon(\sigma) \subset T(\sigma)$ системой неравенств

$$\sum_{i=1}^d a_{i,k} z_i \leq b_k - \varepsilon, \quad k = 1, 2, \dots, K. \tag{11}$$

Будем говорить, что подпоследовательность $x_l, x_{l+1}, \dots, x_{l'}$ удовлетворяет (9), если ограничения (9) выполнены для всех $t = l, l+1, \dots, l' + w - 1$ при подходящем доопределении последовательности векторами $x_{l'+1}, \dots, x_{l'+w-1}$.

Далее через $\|\cdot\|$ обозначается l_∞ -норма в векторном пространстве (максимум модуля координат вектора), а через $\langle \cdot, \cdot \rangle$ – евклидово скалярное произведение векторов. Нам понадобится простой факт о скорости приближения к заранее известному решению задачи (9).

Лемма 1. Пусть $u \in T^\varepsilon(\sigma)$, $v \in T(\sigma)$ и $0 < \varepsilon < \|v - u\|$. Тогда на отрезке $[u, v]$ найдется удовлетворяющая системе неравенств σ (9) ширины w последовательность $\{x_i\}$ с началом $x_0 = x_1 = \dots = x_{w-1} = u$, сходящаяся к точке v со скоростью

$$\|v - x_i\| \leq (1 - \varepsilon \Delta)^{i/w-1} \|v - u\|, \quad \Delta = \|v - u\|^{-1}. \tag{12}$$

Доказательство. Определим $\varepsilon_i = \varepsilon(1 - \varepsilon \Delta)^i$. При $i \geq 1$ и $j = 0, 1, \dots, w - 1$ положим

$$x_{i+w-j} = x_{(i-1)w+j} + \varepsilon_{i-1} \Delta (v - u). \tag{13}$$

По построению

$$v - x_{lw+j} = [1 - (\epsilon_0 + \dots + \epsilon_{i-1})\Delta](v - u) = (1 - \epsilon\Delta)^i(v - u). \tag{14}$$

Таким образом, оценка (12) выполняется. Осталось проверить, что построенная последовательность удовлетворяет системе σ .

В силу $x_{lw} = x_{lw+1} = \dots = x_{lw+w-1} \in [u, v] \subset T(\sigma)$ неравенства (9) выполнены при всех $t = lw$, где $l \in \mathbb{N}$.

Покажем, что $x_{lw} \in T^{\epsilon_l}(\sigma)$. При $l = 0$ это гарантировано условием леммы. Обозначим $a_k = (a_{1,k}, \dots, a_{d,k})$. Тогда при $l > 1$ и при любом $k = 1, 2, \dots, K$, согласно (14), имеет место

$$\langle a_k, x_{lw} \rangle = \langle a_k, (1 - \epsilon_l/\epsilon)v + (\epsilon_l/\epsilon)u \rangle \leq (1 - \epsilon_l/\epsilon)b_k + (\epsilon_l/\epsilon)(b_k - \epsilon) = b_k - \epsilon_l$$

в силу линейности скалярного произведения.

Пусть $lw < t < (l + 1)w$. В этом случае в левых частях неравенств (9) используются только координаты векторов x_{lw} и $x_{(l+1)w}$. Тогда с помощью (13) и с учетом $x_{lw} \in T^{\epsilon_l}(\sigma)$ при любом $k = 1, 2, \dots, K$ получаем

$$\sum_{i=1}^d \sum_{j=0}^{w-1} a_{i,j,k} x_{t+j,i} \leq \sum_{i=1}^d \sum_{j=0}^{w-1} a_{i,j,k} x_{lw,i} + \epsilon_l \sum_{i=1}^d \sum_{j=0}^{w-1} |a_{i,j,k}| \leq (b_k - \epsilon_l) + \epsilon_l.$$

Следовательно, построенная последовательность удовлетворяет системе неравенств σ .

Сформулируем основной результат раздела. Обозначим

$$\pi(a) = \lim_{r \rightarrow \infty} P(r + a) \times 2^{-r}. \tag{15}$$

В частности, $\pi(a) = \frac{17}{14}$ при $0 \leq a < \frac{1}{2}$ и $\pi(a) = \frac{12}{7}$ при $\frac{1}{2} \leq a < 1$, согласно (5).

Лемма 2. Пусть для некоторого набора $y_1, \dots, y_d \in \mathbb{R}$, где $0 \leq y_1 < \dots < y_d < 1$, имеется стратегия сравнений глубины h и ширины w , задающая систему ограничений σ (9) на $x_{r,i}$, где $x_{r,i} \times 2^r$ — нижняя оценка длины интервала, вставка упорядоченной пары в который выполняется с удельной сложностью $2(r + y_i)$. Пусть $v = (v_1, \dots, v_d) \in T(\sigma)$, $u = (u_1, \dots, u_d) \in T^\epsilon(\sigma)$, где $0 < \epsilon < \|v - u\|$, причем $u_i < v_i$ и $u_i \leq \pi(y_i)$ для всех $i = 1, 2, \dots, d$. Обозначим $u_{\min} = \min\{u_i\}$. Тогда для любого $r \in \mathbb{N}$ и любых $m \in \mathbb{N}$, $\phi > 0$, удовлетворяющих условию

$$m \leq \min\{u_{\min}, \epsilon/2\} \times 2^{r-\phi-1}, \tag{16}$$

выполняется

$$P_m \left[r + y_i + \frac{(r + 2)\phi \times 2^\phi}{m} \right] \geq v_i \times 2^r - \frac{R}{\epsilon} m \times 2^{\phi+2} - R \times 2^r e^{-\frac{\epsilon}{2R} \left[\frac{\phi}{w(d+1)(h+1)} - 2 \right]}, \tag{17}$$

где $i = 1, 2, \dots, d$ и $R = \|v - u\|$.

Доказательство. Зададимся некоторым значением r и допустимой парой параметров m и ϕ .

Положим $L = \left\lfloor \frac{\phi}{(h + 1)(d + 1)} \right\rfloor$ и $q = r - L$. Из (16) и (15) вытекает $m \leq \frac{6}{7} \times 2^{r-\phi}$, откуда $r > \phi \geq L$. Следовательно, $q > 0$.

Пусть $\delta \in (0, \epsilon)$ — параметр, который будет выбран позже. На отрезке $[u, v]$ выберем точку $v' = (1 - \delta/\epsilon)v + (\delta/\epsilon)u$. То, что $v' \in T^\delta(\sigma)$, проверяется рассуждением из леммы 1: при любом $k = 1, 2, \dots, K$

$$\langle a_k, v' \rangle = \langle a_k, (1 - \delta/\epsilon)v + (\delta/\epsilon)u \rangle \leq (1 - \delta/\epsilon)b_k + (\delta/\epsilon)(b_k - \epsilon) = b_k - \delta.$$

По определению ширина системы ограничений не превосходит ширину соответствующей ей стратегии; можно считать, что обе величины равны w . Рассмотрим σ' — систему, получающуюся из σ вычитанием δ из правых частей неравенств (9). Тогда имеем $v' \in T(\sigma')$ и $u \in T^{\epsilon-\delta}(\sigma')$. Приме-

няя лемму 1 к системе σ' , построим последовательность $\{x_l = (x_{l,1}, \dots, x_{l,d})\}$, сходящуюся к v' , с началом $x_q = \dots = x_{q+w-1} = u$.

Обозначим $X_{l,i} = x_{q+l,i} \times 2^{q+l}$. По условию заданная стратегия сравнений обеспечивает удельную сложность $2(q+l+y_i)$ вставки пары в интервал длины $X_{l,i}$. Покажем индукцией по $dl+i$, что для вставки m пар в интервал длины

$$X'_{l,i} = X_{l,i} - 2^{(dl+i)h} \times 2m \quad (18)$$

можно построить алгоритм той же удельной сложности $2(q+l+y_i)$ с $c_{dl+i} \leq (dl+i) \times 2^{(dl+i)(h+1)}$ контейнерами. (Напомним, что под интервалом длины $X'_{l,i}$ понимается интервал длины $\lceil X'_{l,i} \rceil$.)

Убедимся, что $X'_{l,i} > 0$. Действительно, согласно (16),

$$X_{l,i} \geq u_i \times 2^{q+l} \geq m \times 2^{\varphi+1-r+q+l} \geq 2m \times 2^{(h+1)(d+1)L+l-L} > 2m \times 2^{(d+1)Lh}.$$

Теперь проверим, что запас по длине и число контейнеров достаточны для вставки, затем обеспечим выполнение ограничений (9) для подпоследовательности векторов x'_{q+l} , $0 \leq l \leq L$, с компонентами $x'_{q+l,i} = X'_{l,i} \times 2^{-(q+l)}$.

При $0 \leq l \leq w-1$ (база индукции) требуемое гарантируется условиями леммы на u_i : длины интервалов находятся в области применения оценок (5). В этом случае контейнеры не нужны. Для вставки m пар требуется запас длины $2m-2$ по отношению к (5) с возможными дополнительными поправками: 1 на округление длины интервала до $\lceil X'_{0,i} \rceil$ и 1 на ошибку (15) при предельном переходе по отношению к (5). Поэтому взятый в (18) запас минимум в $2m$ достаточен.

При $l \geq w$, отталкиваясь от разбиения интервала длины $X_{l,i}$ на подынтервалы в соответствии со стратегией, построим разбиение интервала длины $X'_{l,i}$. По существу задача состоит в том, чтобы распределить запас длины $2^{(dl+i)h} \times 2m$ между подынтервалами. Будем передвигаться по дереву сравнений от корня к произвольному листу. Пусть в очередной вершине выполняется сравнение с элементом³ α главной цепочки, и (α', α'') – минимальный интервал, включающий α , границы которого либо являются элементами предшествующих сравнений, либо границами исходного интервала. Тогда запас длины интервала (α', α'') поделим поровну между интервалами (α', α) и (α, α'') . В итоге, поскольку дерево сравнений имеет глубину h , любой из подынтервалов, образованный точками сравнений, получит запас длины не менее $2^{(dl+i-1)h} \times 2m$ – по индуктивному предположению, достаточный запас для вставки пары с удельной сложностью $2(q+l'+y_j)$ или одиночного элемента – с удельной сложностью⁴ $q+l'+y_j+1/2$, где $l' < l$ или $j < i$. Разумеется, запас выбирается в пределах длины подынтервала, чтобы она оставалась положительной. Также заметим, что терминальные интервалы в алгоритме имеют удельную сложность вставки пары не менее $q+l-w \geq q$, поэтому рекурсивный вызов алгоритма (вставки в терминальный интервал) возможен. Формально не исключается случай, когда стратегия предполагает вставку одиночного элемента во внутренний подынтервал с удельной сложностью не менее $q+l+y_i+1/2$. В этой ситуации просто вызывается алгоритм вставки в исходный интервал стартовой длины $X'_{l,i}$.

С использованием разбиения интервала длины $X'_{l,i}$, разбиение интервала длины $\lceil X'_{l,i} \rceil$ строится простым округлением рангов всех внутренних элементов разбиения до ближайших целых чисел сверху. При этом длины любых подынтервалов исходного разбиения, в том числе терминальных, изменяются менее чем на 1, т.е. превращаются в округления до ближайших целых сверху или снизу. Таким образом, учет округлений не приводит к изменению оценки удельной сложности алгоритма.

³ Поскольку стратегия оперирует с нецелыми длинами интервалов, под элементом в этом абзаце понимается условная точка разбиения интервала.

⁴ Под удельной сложностью вставки одиночного элемента понимается приходящаяся на него удельная сложность вставки в рамках пары.

Оценим достаточное число контейнеров. Дерево сравнений имеет не более 2^h листьев, каждому соответствует до двух терминальных интервалов, вставка в которые по предположению индукции выполняется алгоритмами, использующими не более $2c_{dl+i-1}$ контейнеров. Еще не более $2^{h+1} + 1$ контейнеров может понадобиться для обслуживания непосредственно терминальных интервалов и исходного интервала. Получаем

$$c_{dl+i} \leq 2^{h+1}[(dl + i - 1) \times 2^{(dl+i-1)(h+1)}] + 2^{h+1} + 1 \leq (dl + i) \times 2^{(dl+i)(h+1)}.$$

Теперь проверим выполнение ограничений σ для $x'_{l,i}$. Формально положим $x'_{l,i} = x_{j,i}$ при $j > r$. Определим $\delta = 2^{\varphi-r+1}m$. По условию (16) имеет место $\delta < \varepsilon/2$. Кроме того, в силу (18) и определенных параметров φ и δ выполнено $x'_{l,i} - x_{j,i} \in (-\delta, 0]$ для всех $j \geq q$. Значит, при всех $t \geq q$ и любом $k = 1, 2, \dots, K$ получаем

$$\sum_{i=1}^d \sum_{j=0}^{w-1} a_{i,j,k} x'_{t+i,i} \leq \sum_{i=1}^d \sum_{j=0}^{w-1} a_{i,j,k} x_{t+j,i} + \delta \sum_{i=1}^d \sum_{j=0}^{w-1} |a_{i,j,k}| \leq (b_k - \delta) + \delta.$$

Таким образом, последовательность $\{x'_l = (x'_{l,1}, \dots, x'_{l,d})\}$ удовлетворяет системе ограничений σ при $t \geq q$. Тем самым доказан индуктивный переход: вставка в интервал длины $X'_{L,i}$ действительно выполняется с удельной сложностью $2(r + y_i)$ сравнений.

Остается проверить оценку (17). Выбором параметра L обеспечено

$$c_{dl+i} \leq (d + 1)L \times 2^{(d+1)(h+1)L} \leq \varphi \times 2^\varphi. \tag{19}$$

Из леммы 1 следует

$$\begin{aligned} |x'_{r,i} - v_i| &\leq |x'_{r,i} - x_{r,i}| + |x_{r,i} - v'_i| + |v'_i - v_i| \leq \delta + (1 - (\varepsilon - \delta)/R)^{L/w-1} R + (\delta/\varepsilon)|v_i - u_i| \leq \delta + \\ &+ e^{-(\varepsilon-\delta)(L/w-1)/R} R + (\delta/\varepsilon)R \leq \delta + e^{-\frac{\varepsilon}{2R} \left[\frac{\varphi}{w(d+1)(h+1)} - 2 \right]} R + (\delta/\varepsilon)R < \frac{2R}{\varepsilon} m \times 2^{\varphi-r+1} + e^{-\frac{\varepsilon}{2R} \left[\frac{\varphi}{w(d+1)(h+1)} - 2 \right]} R, \end{aligned} \tag{20}$$

где также использовалось известное неравенство $(1 - x)^{1/x} \leq e^{-1}$ при $x \in (0, 1)$. Теперь (17) получается подстановкой (19) и (20) в оценку (8), в которой можно положить $n + 2m \leq X_{L,i} = x_{r,i} \times 2^r < 2^{r+2}$, поскольку из теоретико-информационных соображений $x_{r,i} \leq 2^{y_i+1/2} < 4$.

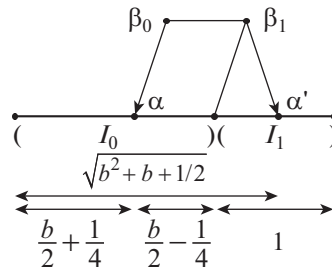
Оценки в обеих леммах весьма грубы: точность приносится в жертву простоте или общности изложения. В частности, существенно меньшее число контейнеров требуется конкретной стратегии, к которой будет применяться лемма 2.

5. УНИВЕРСАЛЬНАЯ СТРАТЕГИЯ

В этом разделе строится стратегия сравнений для доказательства асимптотической оценки высокой точности. Известно (и вполне очевидно), что для построения оптимального алгоритма сортировки некоторого частично упорядоченного множества желательно, чтобы при каждом сравнении множество возможных доупорядочиваний разбивалось на две примерно равные части в зависимости от результата сравнения. Такой ход мысли ведет к градиентному методу: сначала выполняем некоторое число сравнений, руководствуясь делением множества исходов как можно точнее пополам; затем полученный частичный порядок обрабатываем каким-то простым способом.

Теоретико-информационная нижняя оценка сложности вставки пары в интервал длины $n - 1$ составляет $\log[n(n - 1)/2] < 2(\log n - 1/2)$. Представим идеальный случай, предположив, что с удельной сложностью $2a$ пару можно вставить в интервал длины $Y(a) = (1 - \lambda) \times 2^{a+1/2}$ при любом достаточно большом a , где $\lambda \geq 0$ – не зависящий от a параметр, который будет определен позднее.

Рассмотрим одну из градиентных стратегий для алгоритма вставки пары $\beta_0 < \beta_1$ в интервал длины $Y(a)$. Стратегия характеризуется параметром $s \in \mathbb{N}$. Сразу заметим, что при построении стратегии длины интервалов считаются действительными числами: необходимость округления до целых учитывается при переходе от стратегии к алгоритму леммой 2.



Фиг. 2. Оптимальные варианты для очередного сравнения.

Сначала отдельно рассмотрим задачу выбора очередного элемента для сравнения. Пусть в результате серии сравнений частичный порядок таков, что элемент β_1 принадлежит интервалу I_1 , а β_0 принадлежит интервалу $I_0 \cup I_1$, где $I_0 \cap I_1 = \emptyset$ и $|I_0| = b|I_1|$. Тогда наилучший выбор для сравнения с β_0 – это элемент⁵ α , имеющий ранг $(b/2 + 1/4)|I_1|$, считая от левого края интервала I_0 . (Если $b > 1/2$, то $\alpha \in I_0$.) А наилучший выбор для сравнения с β_1 – это элемент α' ранга $\sqrt{b^2 + b + 1/2}|I_1|$ (фиг. 2). Вообще, если бы мы хотели разбить множество исходов в отношении $x : (1 - x)$, то для сравнения с β_0 и β_1 нужно было бы выбрать соответственно элементы ранга $x(b + 1/2)|I_1|$ или ранга $\sqrt{b^2 + x(2b + 1)}|I_1|$.

Исходное разбиение. Не ограничивая общности, считаем, что первое сравнение имеет вид $\beta_1 ? \alpha_1$, тогда в качестве α_1 следует взять элемент ранга $Y(a - 1/2) = (1 - \lambda) \times 2^a$. Если $\beta_1 < \alpha_1$, то вызывается алгоритм вставки пары в интервал $(-\infty, \alpha_1)$ длины $Y(a - 1/2)$. Иначе, выполняем сравнение $\beta_0 ? \alpha_2$, где α_2 – элемент оптимального ранга $(1 - \lambda) \frac{\sqrt{2} + 1}{4} \times 2^a$. Причина, по которой второе сравнение выполняется с элементом β_0 , будет разъяснена чуть позже.

Если $\beta_0 < \alpha_2$, то элемент β_0 вставляется в интервал $(-\infty, \alpha_2)$ с удельной сложностью $a - 2 + \log(\sqrt{2} + 1)$, а элемент β_1 вставляется в интервал $(\alpha_1, +\infty)$ длины $(1 - \lambda)(\sqrt{2} - 1) \times 2^a$ с удельной сложностью $a + \log(\sqrt{2} - 1)$. Следовательно, в этом случае вставка пары выполняется за $2a$ удельных сравнений.

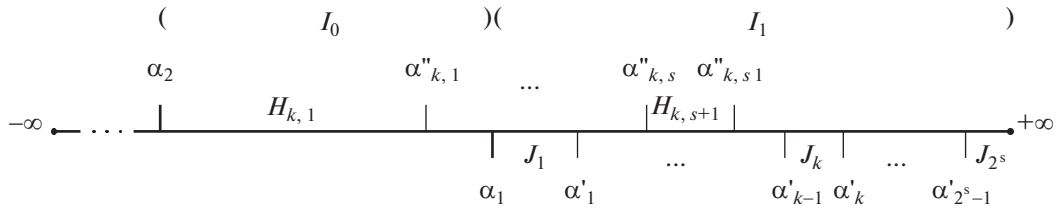
Если же $\beta_0 > \alpha_2$, то $\beta_1 \in I_1$ и $\beta_0 \in I_0 \cup I_1$, где $I_1 = (\alpha_1, +\infty)$ и $I_0 = (\alpha_2, \alpha_1)$. Пусть $b = |I_0|/|I_1| = \frac{2\sqrt{2} + 1}{4}$. С помощью s сравнений, разбивая каждый раз множество исходов пополам, мы локализуем β_1 в одном из 2^s подынтервалов интервала I_1 . В соответствии с приведенными выше формулами границами этих подынтервалов являются элементы α'_k ранга $b_k \cdot |I_1|$, $b_k = \sqrt{b^2 + k(2b + 1)} \times 2^{-s}$, $k = 0, 1, \dots, 2^s - 1$, и условная точка $\alpha'_{2^s} = +\infty$. Обозначим подынтервал $(\alpha'_{k-1}, \alpha'_k)$ через J_k . Общая схема разбиения показана на фиг. 3.

Пусть элемент β_1 определен в интервал J_k . Следующий этап – вставка элемента β_0 . По очереди сравниваем его с элементами $\alpha''_{k,1}, \dots, \alpha''_{k,s+1}$, определяемыми следующим образом. Положим $\alpha''_{k,0} = \alpha_2$. Обозначим через $H_{k,j}$ интервал $(\alpha''_{k,j-1}, \alpha''_{k,j})$. Длина интервала $H_{k,j}$ выбирается равной

$$|H_{k,j}| = (1 - \lambda) \times 2^{2a - s - 2 - j - \log\lfloor J_k / (1 - \lambda) \rfloor} \tag{21}$$

Если $\beta_0 > \alpha''_{k,1}$, то выполняем сравнение $\beta_0 ? \alpha''_{k,2}$; если $\beta_0 > \alpha''_{k,2}$, то сравниваем β_0 с $\alpha''_{k,3}$ и т.д. Как только $\beta_0 < \alpha''_{k,j}$, элементы β_0 и β_1 вставляются независимо в интервалы $H_{k,j}$ и J_k с суммарной удельной сложностью $2a - s - 2 - j$. В этом случае общая удельная сложность вставки пары составит $2a$.

⁵ Здесь и ниже под элементом обычно понимается точка разбиения интервала.



Фиг. 3. Универсальное разбиение на интервалы.

Наконец, если $\beta_0 > \alpha''_{k,s+1}$, то оба элемента β_0, β_1 принадлежат интервалу $(\alpha''_{k,s+1}, \alpha'_k)$. Хотелось бы вставить пару в этот интервал ценой $2a - 2s - 3$ сравнений. Это было бы возможно в случае, когда, скажем, $\alpha''_{k,s+1} = \alpha_{k-1}$. Но равенство, вообще говоря, не выполняется, и интервал оказывается длиннее требуемого. Оценим превышение длины. Интервал J_k имеет длину

$$|J_k| = (b_k - b_{k-1})|I_1| = \frac{2b + 1}{2^s(b_k + b_{k-1})}|I_1| = (1 - \lambda) \frac{\sqrt{2} + 1}{2^{s+1}(b_k + b_{k-1})} \times 2^a. \tag{22}$$

Тогда из (21) следует, что

$$|H_{k,j}| = (1 - \lambda) \times 2^{a-1-j-\log(\sqrt{2}+1)+\log(b_k+b_{k-1})} = \frac{b_k + b_{k-1}}{2^{j+1}}|I_1|. \tag{23}$$

Значит, сумма длин интервалов $H_{k,j}$ при фиксированном k равна

$$\sum_{j=1}^{s+1} |H_{k,j}| = (1 - 2^{-s-1}) \frac{b_k + b_{k-1}}{2} |I_1|. \tag{24}$$

Получаем

$$\begin{aligned} |(\alpha''_{k,s+1}, \alpha'_k)| &= b_k |I_1| - \sum_{j=1}^{s+1} |H_{k,j}| = \left[b_k - (1 - 2^{-s-1})(b_k + b_{k-1})/2 \right] |I_1| = \\ &= \left(\frac{b_k - b_{k-1}}{2} + \frac{b_k + b_{k-1}}{2^{s+2}} \right) |I_1| < \left(\frac{2b + 1}{2^{s+1} \times 2b} + \frac{b + 1}{2^{s+1}} \right) |I_1| < \frac{7}{2^{s+2}} |I_1|. \end{aligned}$$

Следовательно,

$$|(\alpha''_{k,s+1}, \alpha'_k)| - Y(a - s - 3/2) < \frac{7}{2^{s+2}} |I_1| - (1 - \lambda) \times 2^{a-s-1} = \frac{7 - 2(\sqrt{2} + 1)}{2^{s+2}} |I_1| < \frac{\ln 2}{2^s} |I_1|. \tag{25}$$

Полученная оценка понадобится чуть позже. Теперь мы модифицируем описанную выше стратегию сравнений, позволив некоторое отступление от идеала, т.е. деления множества исходов ровно пополам. А именно, мы изменим распределение длин подынтервалов J_k внутри интервала I_1 , уменьшив (оценочную) сложность вставки элементов в каждый из них, тем самым увеличив допустимую сложность вставки в интервалы $H_{k,j}$ и удлинив их. Для этого у нас имеется резерв – использовать для сложности вставки элемента простую оценку $Q(a)$ вместо $Y(a - 1/2)$, которая лучше, когда значение a близко снизу к целому числу⁶. Длина интервала I_1 не будет изменена.

Заметим, что длины интервалов J_k с ростом k изменяются в пределах от $\frac{2b+1}{2b} \times 2^{-s} |I_1|$ до $\frac{2b+1}{2b+2} \times 2^{-s} |I_1|$ (см. (22)). В силу $b < 1$ выполнено $\left[\frac{2}{3}, \frac{4}{3} \right] \subset \left[1 - \frac{1}{2b+1}, 1 + \frac{1}{2b+1} \right]$, поэтому при любом (достаточно большом) a часть интервалов J_k имеет длину, близкую к степени двойки. Выполненное вторым сравнением $\beta_0 ? \alpha_2$ преследовало именно цель уменьшения показателя b до величины меньше 1.

Коррекция разбиения. На этом шаге мы вносим поправки в длины интервалов J_k и $H_{k,j}$, корректируя стратегию, и одновременно дискретизируем задачу (новые интервалы будут обозна-

⁶ Едва ли можно построить эффективный алгоритм вставки, не прибегая к оценкам типа $Q(a)$, что видно на примерах алгоритмов из [10], [11] и простого алгоритма из разд. 3.

чаться как J_k^* и $H_{k,j}^*$). Введем параметр дискретизации $d \in 2\mathbb{N}$ и положим $y_i = i/d$, $i = 0, 1, \dots, d-1$. По-прежнему мы ищем оценку длины интервала, достаточной для вставки пары с удельной сложностью $2(r + y_i)$, в виде $Y(r + y_i) = (1 - \lambda) \times 2^{r+y_i+1/2}$.

Задавшись некоторым $a = r + y_i$, $r \in \mathbb{N}$, опишем модификацию исходного разбиения. Четность параметра d позволяет выбрать элемент α_1 для первого сравнения так же, как и в идеальной схеме. Но сложность вставки элемента в интервал $I_1 = (\alpha_1, +\infty)$ теперь приходится оценить как $a + \lceil d \log(\sqrt{2} - 1) \rceil / d < a + \log(\sqrt{2} - 1) + 1/d$. Как следствие, вместо α_2 приходится выбирать элемент α_2^* ранга не выше $(1 - \lambda) \times 2^{a-2-\lceil d \log(\sqrt{2}-1) \rceil / d}$. Руководствуясь соображением последующего применения леммы 2, создадим дополнительный запас сложности $1/d$ на втором сравнении и выберем в качестве α_2^* элемент чуть меньшего ранга:

$$(1 - \lambda) \times 2^{a-2-\lceil d \log(\sqrt{2}-1) \rceil / d - 1/d} > (1 - \lambda) \frac{\sqrt{2} + 1}{4} \times 2^{a-2/d}. \quad (26)$$

Тогда, используя неравенство $e^x \geq 1 + x$, получаем

$$|I_0^*| = |(\alpha_2^*, \alpha_1)| \leq (1 - \lambda) \times 2^a \left(1 - \frac{\sqrt{2} + 1}{4} \times 2^{-2/d} \right) \leq (1 - \lambda) \times 2^a \left(\frac{3 - \sqrt{2}}{4} + \frac{\sqrt{2} + 1}{2d} \ln 2 \right).$$

Следовательно,

$$0 \leq |I_0^*| - |I_0| \leq \frac{(3 + 2\sqrt{2})}{2d} \ln 2 |I_1|. \quad (27)$$

При d достаточно большом, $|I_0^*| < |I_1|$.

Выполним перераспределение интервалов J_k внутри I_1 . Введем параметр $\mu > 0$. Потребуем, чтобы удельная сложность вставки элемента в J_k^* была меньше сложности вставки в J_k в идеальной стратегии хотя бы на $\left(2 - \frac{k}{2^s}\right)\mu$ сравнений, $k = 1, 2, \dots, 2^s$. Напомним, что в идеальной стратегии сложность оценивается как $\log[|J_k|/(1 - \lambda)]$. Прибегать к неравномерности сужения вынуждает эффект смещения правых границ интервалов J_k . При равномерном сужении в $1 + \mu$ раз, ввиду $|I_0| < |I_1|$, компенсация этого смещения, вообще говоря, была бы затруднена при малых k : сумма длин интервалов $H_{k,j}$ выросла бы примерно на $\mu |I_0|$, тогда как точка α'_{k+1} (правая граница интервала J_k) могла бы сместиться вправо приблизительно на $\mu |I_1|$. Поэтому для интервалов с большими номерами выбирается меньший коэффициент сужения.

Сужение. Мы собираемся сузить почти все интервалы J_k , но это сужение будет компенсировано расширением небольшого числа интервалов, длины которых близки к степеням двойки, с опорой на простую оценку (4). Сначала оценим сверху величину θ_p общего сужения интервалов с номерами от p до 2^s . Перед этим заметим, что для величины b_k выполняются простые соотношения (проверяются возведением в квадрат)

$$b + \frac{k}{2^s} b \leq b_k \leq b + \frac{k}{2^s} \frac{2b + 1}{2b^2}. \quad (28)$$

С учетом дискретизации длину суженного интервала J_k^* можно выбрать в пределах

$$2^{-\left(2 - \frac{k}{2^s}\right)\mu - 1/d} |J_k| \leq |J_k^*| \leq 2^{-\left(2 - \frac{k}{2^s}\right)\mu} |J_k|. \quad (29)$$

Исходя из пессимистического предположения, что все интервалы сужаются, снова используя неравенство $e^x \geq 1 + x$, а также (22) и (28), получаем

$$\begin{aligned} \theta_p &\leq \sum_{k=p}^{2^s} (|J_k| - |J_k^*|) \leq \sum_{k=p}^{2^s} |J_k| \left[1 - 2^{-\left(2-\frac{k}{2^s}\right)\mu - \frac{1}{d}} \right] \leq \sum_{k=p}^{2^s} |J_k| [(2 - k2^{-s})\mu + 1/d] \ln 2 = \left(2\mu + \frac{1}{d}\right) \times \\ &\times (b + 1 - b_{p-1}) \ln 2 |I_1| - \mu 2^{-s} \ln 2 \sum_{k=p}^{2^s} k |J_k| = \left(2\mu + \frac{1}{d}\right) (b + 1 - b_{p-1}) \ln 2 |I_1| - \mu 2^{-2s} (2b + 1) \ln 2 \times \\ &\times \sum_{k=p}^{2^s} \frac{k}{b_k + b_{k-1}} |I_1| \leq \left[2\mu \left(1 - \frac{p-1}{2^s} b\right) - \mu 2^{-2s} \frac{2b+1}{2(b+1)} \frac{2^{2s} - p^2}{2} + \frac{1}{d} \right] \ln 2 |I_1| = \\ &= \left[\left[2 - 2b \frac{p-1}{2^s} - \frac{2b+1}{4(b+1)} \left(1 - \frac{p^2}{2^{2s}}\right) \right] \mu + \frac{1}{d} \right] \ln 2 |I_1|. \end{aligned}$$

В частности, для суммарного сужения всех интервалов J_k имеем оценку

$$\theta_1 \leq (2\mu + 1/d) \ln 2 |I_1|, \tag{30}$$

а при $p \geq 2$ – оценку

$$\theta_p \leq \left[\left[2 - \frac{2b+1}{4(b+1)} \right] \mu + \frac{1}{d} \right] \ln 2 |I_1|, \tag{31}$$

поскольку $p^2/2^s \leq p \leq 2(p-1)$ и $\frac{2b+1}{4(b+1)} < b = \frac{2\sqrt{2}+1}{4}$.

Расширение. Теперь оценим снизу величину возможной компенсации. Если

$$(1 - \lambda) \times 2^{l+2\mu} \leq |J_k| < 2^l, \quad l \in \mathbb{N},$$

то сложность вставки элемента пары в интервал J_k в идеальной стратегии пока оценивалась как минимум в $l + 2\mu$. Тем не менее длину интервала можно увеличить до 2^l , при этом уменьшив оценку сложности по меньшей мере на 2μ , что и требуется.

Сначала проверим, что если λ и μ не слишком велики, то при некотором $l \in \mathbb{N}$ справедливо

$$|J_{2^s}| \leq (1 - \lambda) \times 2^{l+2\mu} < 2^l \leq |J_1|. \tag{32}$$

При $\mu \ll \lambda$ для этого достаточно выполнения условия $|J_1|/|J_{2^s}| \geq \frac{2}{1-\lambda}$. Используя (22) и (28), выводим

$$\frac{|J_1|}{|J_{2^s}|} = \frac{b_{2^s} + b_{2^s-1}}{b_1 + b_0} \geq \frac{2b + 2 - \frac{b}{2^s}}{2b + \frac{2b+1}{2b^2 \times 2^s}} \geq \frac{b+1}{b} \left[1 - \frac{b}{2(b+1) \times 2^s} \right] \left(1 - \frac{2b+1}{4b^3 \times 2^s} \right) > \frac{b+1}{b} (1 - 2^{1-s}),$$

где также использовались простые неравенства $\frac{1}{1+x} \geq 1-x$ и $(1-x)(1-y) \geq 1-x-y$, справедливые при $x, y \geq 0$. Подстановкой численных значений легко проверяется, что полученная оценка превосходит $\frac{2}{1-\lambda}$, скажем, при любых $s \geq 10$ и $\lambda \leq 1/50$.

Итак, некоторый отрезок $g = [(1 - \lambda) \times 2^{l+2\mu}, 2^l]$ лежит внутри отрезка $[|J_{2^s}|, |J_1|]$. Оценим возможное увеличение длины только для тех интервалов J_k , длины которых оказались в g . Напомним, что длины интервалов монотонно убывают с ростом k (см. (22)). Длину отрезка g оценим снизу с помощью (22) и справедливого при $0 \leq x \leq 1$ неравенства $2^x \leq 1 + x$ как

$$[1 - (1 - \lambda)2^{2\mu}] \times 2^l \geq [1 - (1 - \lambda)(1 + 2\mu)] |J_{2^s}| \geq L = (\lambda - 2\mu) \frac{2b+1}{2(b+1) \times 2^s} |I_1|. \tag{33}$$

Теперь оценим сверху разность длин соседних интервалов:

$$\frac{2^s}{(2b+1)|I_1|} (|J_k| - |J_{k+1}|) = \frac{1}{b_k + b_{k-1}} - \frac{1}{b_{k+1} + b_k} \leq \frac{b_{k+1} - b_{k-1}}{4b^2} = \frac{2(2b+1)}{4b^2(b_{k+1} + b_{k-1}) \times 2^s} \leq \frac{2b+1}{4b^3 \times 2^s}.$$

Следовательно, при любом k

$$|J_k| - |J_{k+1}| \leq \Delta = \frac{(2b+1)^2}{4b^3 \times 2^{2s}} |I_1|. \quad (34)$$

Легко проверить, что если множество точек делит отрезок длины L на подотрезки с длинами не более Δ , то сумма расстояний от этих точек до границы отрезка не меньше $\frac{L(L-\Delta)}{2\Delta}$. Подставляя в эту формулу значения L и Δ , приходим к оценке

$$\begin{aligned} \frac{L(L-\Delta)}{2\Delta} &\geq \left[(\lambda - 2\mu)^2 \frac{b^3}{2(b+1)^2} - (\lambda - 2\mu) \frac{2b+1}{4(b+1) \times 2^s} \right] |I_1| = \\ &= (\lambda - 2\mu) \left[\lambda - 2\mu - \frac{(2b+1)(b+1)}{2b^3 \times 2^s} \right] \frac{b^3}{2(b+1)^2} |I_1| > (\lambda - 2\mu)(\lambda - 2\mu - 2^{2-s}) \frac{b^3}{2(b+1)^2} |I_1|. \end{aligned}$$

Так мы оценили снизу максимум суммарного возможного расширения интервалов J_k с длинами из отрезка g (до длины 2^l).

Компенсация сужения остальных интервалов J_k осуществима при $\frac{L(L-\Delta)}{2\Delta} \geq \theta_1$. Используя (30), достаточное для этого условие можно теперь записать как

$$(\lambda - 2\mu)(\lambda - 2\mu - 2^{2-s}) \frac{b^3}{2(b+1)^2} \geq \left(2 + \frac{1}{d}\right) \ln 2\mu. \quad (35)$$

Пока ограничимся тривиальным замечанием: при достаточно большом s можно задать параметры в виде⁷ $d \asymp 2^s$, $\lambda \asymp 2^{-s/2}$, $\mu \asymp 2^{-s}$ так, что (35) будет удовлетворено.

Сведение. Пришло время проверить, позволяют ли предпринятые меры компенсировать дефицит длины интервалов $H_{k,j}$, отраженный в соотношении (25). Обратим внимание на изменение условий, в которых выполнялась оценка (25). Для величины $Y(a-s-3/2)$ используется прежняя оценка, но при этом произошло удлинение интервала I_0 (см. (27)), а правая граница интервала J_k могла сместиться правее на величину, оцененную сверху как θ_{k+1} (см. (31)).

Таким образом, мы требуем, чтобы при любом k прирост суммарной длины интервалов $H_{k,j}$ составил не менее

$$|I_0^*| - |I_0| + \theta_{k+1} + \frac{\ln 2}{2^s} |I_1| + \delta |I_1|, \quad (36)$$

где $\delta |I_1|$ – дополнительный запас, который потребуется позже для применения леммы 2.

По построению, $|H_{k,j}^*| \geq |H_{k,j}| \times 2^{\left(\frac{2-k}{2^s}\right)\mu - \frac{1}{d}}$ с учетом поправки на дискретизацию. Поэтому, подставляя (24) и используя (28), получаем

$$\begin{aligned} \sum_{j=1}^{s+1} (|H_{k,j}^*| - |H_{k,j}|) &\geq \left[2^{\left(\frac{2-k}{2^s}\right)\mu - \frac{1}{d}} - 1 \right] \sum_{j=1}^{s+1} |H_{k,j}| \geq \left[\left(2 - \frac{k}{2^s}\right)\mu - \frac{1}{d} \right] \ln 2 \left(1 - \frac{1}{2^{s+1}}\right) \frac{b_k + b_{k-1}}{2} |I_1| \geq \\ &\geq \left(2 - \frac{k}{2^s}\right) \left(1 - \frac{1}{2^{s+1}}\right) \left(1 + \frac{k - \frac{1}{2}}{2^s}\right) \mu b \ln 2 |I_1| - \frac{b+1}{d} \ln 2 |I_1| \geq \\ &\geq \left(2 + \frac{k-2}{2^s} - \frac{k^2}{2^{2s}}\right) \mu b \ln 2 |I_1| - \frac{b+1}{d} \ln 2 |I_1| \geq \left[(2 - 2^{1-s})\mu b - \frac{b+1}{d}\right] \ln 2 |I_1|. \end{aligned}$$

⁷ Символ \asymp означает равенство порядков роста.

Используя эту оценку, а также (27) и (31), достаточное для выполнения (36) условие можно записать как

$$(2 - 2^{1-s})\mu b - \frac{b+1}{d} \geq \frac{(3 + 2\sqrt{2})}{2d} + \left[2 - \frac{2b+1}{4(b+1)} \right] \mu + \frac{1}{d} + \frac{1}{2^s} + \delta \log e. \tag{37}$$

Положим $d = 2^s$ и $\delta = 2^{-s}$. Тогда, чтобы удовлетворить (37), можно выбрать $\mu = 30 \times 2^{-s}$ (принимая, что $s \geq 10$). Условие (35) при этом будет выполнено, например, если $\lambda = 20 \times 2^{-s/2} + 64 \times 2^{-s}$. Обеспечивающее (32) частное условие $\lambda \leq 1/50$ выполнено при любом $s \geq 20$.

Описание стратегии. Фиксируя выбранные значения параметров d, δ, μ, λ , получаем стратегию вставки с системой ограничений, которую обозначим через $\sigma[s]$. Стратегия заключается в том, что элементы главной цепочки, участвующие в сравнениях, разбивают цепочку на интервалы, вставка в которые выполняется с такой сложностью, которая получается в вышеприведенном расчете. Опишем систему ограничений стратегии.

Пусть $p(a)$ является оценкой длины интервала с удельной сложностью вставки пары $2a$ сравнений, а $p_c(J)$ — оценкой длины интервала с удельной сложностью вставки элемента на c выше, что и в некоторый интервал J из построенного ранее разбиения. Для вставки в некоторые интервалы J мы выбираем оценки типа $Q(a)$, а для остальных вставок — оценки типа $p(a)$ при подходящих a .

Первое сравнение, в силу выбора элемента α_1 , не порождает ограничений. Для второго сравнения вводится неравенство

$$p(a) \leq p(a - 1/2) + p_{1/d}(I_1). \tag{38}$$

Здесь мы используем запас сложности, обеспеченный выбором элемента α_2^* (см. (26)), и можем позволить вставку в интервал I_1 с чуть большей удельной сложностью. Это понадобится только для формального удовлетворения условиям применения леммы 2.

Следующие s сравнений, локализуящие β_1 в одном из интервалов J_k^* , относятся к внутренним вершинам дерева сравнений. Для того чтобы обеспечить в дальнейшем вставку элемента β_1 в любой из этих интервалов с заданной сложностью, вводится ограничение

$$p(a) \leq p(a - 1/2) + \sum_{k=1}^{2^s} p_0(J_k^*). \tag{39}$$

Ограничение всего одно, так как длины интервалов $J_k^*, k > 1$, мы можем задать свободно, выбирая подходящим образом пограничные элементы α_k^* , тогда только длина интервала J_1^* будет определяться длиной остальных интервалов.

Длины интервалов $H_{k,j}^*$ тоже устанавливаются свободно (выбором элементов для сравнений), и лишь вставки после $2s + 3$ сравнений порождают ограничения вида

$$p(a) \leq p_0(I_{-1}) + \sum_{j=1}^{s+1} p_0(H_{k,j}^*) + p(a - s - 3/2) + \sum_{i=k+1}^{2^s} p_0(J_i^*), \tag{40}$$

где $I_{-1} = (-\infty, \alpha_2^*)$.

Таким образом, система ограничений $\sigma[s]$ складывается из (38)–(40) при всевозможных $a = r + i/d$. Для перехода к форме записи из определения стратегии следует в указанных неравенствах заменить выражения $p(x)$ и $p_c(J)$ величинами $p_{r,i}$ или числовыми значениями в случаях, когда применяется оценка $Q(a)$.

Глубина стратегии с системой ограничений $\sigma[s]$ равна $2s + 3$. Проверим, что ширина (стратегии и, следовательно, системы $\sigma[s]$) не превосходит $s + 4$. Действительно,

$$\frac{|I_1|}{p(a)} = \sqrt{2} - 1 > \frac{1}{2^2}, \quad \frac{|I_{-1}|}{p(a)} \geq \frac{\sqrt{2} + 1}{4 \times 2^{2/d}} > \frac{1}{2},$$

согласно (26) (напомним, что $p(a) = Y(a)$). Далее, в силу (29) и (22)

$$\frac{|J_k^*|}{p(a)} \geq \frac{|J_k|}{p(a) \times 2^{2\mu+1/d}} \geq \frac{\sqrt{2} + 1}{(b + 1) \times 2^{s+2+2\mu+1/d}} > \frac{1}{2^{s+2}}.$$

Наконец, из (23) следует

$$\frac{|H_{k,s+1}^*|}{p(a)} \geq \frac{|H_{k,s+1}|}{p(a)} \geq \frac{b}{2^{s+1}} \frac{|I_1|}{p(a)} > \frac{1}{2^{s+3}}.$$

Таким образом, если $p(a) = p_{r',i'}$, то при перезаписи (38)–(40) в терминах величин $p_{r,i}$, среди этих величин будут встречаться только такие, для которых $r \geq r' - s - 3$.

Лемма 3. Пусть $s \geq 20$, $d = 2^s$, $\mu = 30 \times 2^{-s}$, $\lambda = 20 \times 2^{-s/2} + 64 \times 2^{-s}$ и $\varepsilon = \frac{1}{25 \times 2^s}$. Пусть также

$y_i = i/d$ и $v_i = (1 - \lambda) \times 2^{y_i+1/2}$. Тогда

(i) $v = (v_0, \dots, v_{d-1}) \in T(\sigma[s])$;

(ii) $u = v/2 \in T^\varepsilon(\sigma[s])$.

Доказательство. Первая часть леммы уже доказана, так как сама стратегия подогнана под заданное решение. Остается проверить (ii).

Напомним, что при переходе к канонической системе (9) мы переписываем неравенства в терминах переменных $x_{r,i} = p_{r,i} \times 2^{-r}$ и выполняем нормировку, а для перехода к приведенной системе (10) производится подстановка $x_{r,i} = z_i$.

Пусть $p(a) = p_{r',i'}$. Выполним предварительную нормировку неравенств (38)–(40) домножением на $2^{-r'}$ и перепишем их в терминах величин $x_{r,i}$. Полученные неравенства обозначим через (38')–(40'). Оценим сумму абсолютных величин коэффициентов при $x_{r,i}$ в каждом из этих неравенств для определения итоговых коэффициентов нормировки.

Вклад от слагаемого $p(a)$ в каждую сумму коэффициентов равен 1. Тогда вклад любого из слагаемых $p(a - 1/2)$, $p_{1/d}(I_1)$, $p_0(I_{-1})$ и $p_0(H_{k,1})$ не выше 1, вклад $p(a - s - 3/2)$ не превосходит 2^{-s-1} , при любом j вклад слагаемого $p_0(H_{k,j+1})$ вдвое меньше, чем у $p_0(H_{k,j})$. Поскольку $|J_k^*| \leq |J_1| \leq 2^{a-s}$, согласно (22), то вклад каждого из слагаемых $p_0(J_k^*)$ не превосходит 2^{-s} . Эти оценки означают, что суммы коэффициентов при $x_{r,i}$ в неравенствах (38'), (39') и (40') не превосходят соответственно 3, 3 и 5.

Теперь оценим запас, с которым вектор u удовлетворяет приведенным неравенствам (38')–(40'). По построению для основного решения v приведенное неравенство (38') выполнялось бы и при замене $p_{1/d}(I_1)$ на $p_0(I_1)$ в (38). Таким образом, правая часть (38) при $p_{r,i} = v_i \times 2^r$ превосходит левую на величину не менее

$$p_{1/d}(I_1) - p_0(I_1) \geq |I_1|(2^{1/d} - 1) \geq \frac{\ln 2}{d}|I_1| = (1 - \lambda)(\sqrt{2} - 1) \ln 2 \times 2^{a-s} > 2^{a-s-2}.$$

После нормировки (делением не более, чем на 3×2^a) и подстановки u вместо v величина запаса в приведенном неравенстве (38') сокращается не более, чем в 6×2^a раз, и остается не меньшей, чем $\frac{1}{24 \times 2^s} > \varepsilon$.

Оценим свободный член, возникающий в (39') из-за того, что для сложности вставки в некоторые интервалы J_k^* используется оценка $Q(a)$. Число таких интервалов не меньше $\lfloor L/\Delta \rfloor$ (см. выше), а применяемая оценка имеет величину $2^l \geq |J_{2^l}| \geq 2^{a-s-2}$, согласно (32) и (22). Поэтому, используя значения L и Δ из (33) и (34), заключаем, что свободный член в правой части (39') не меньше

$$\frac{L - \Delta}{\Delta \times 2^{s+2}} = (\lambda - 2\mu) \frac{b^3}{2(b+1)(2b+1)} - \frac{1}{2^{s+2}} > \frac{1}{2^{s/2}}.$$

При подстановке u вместо v в приведенное неравенство (39') все члены сокращаются вдвое, тогда и свободный член можно уменьшить наполовину, т.е. на $\frac{1}{6 \times 2^{s/2}} > \varepsilon$ с учетом нормировки.

Каждое из неравенств (40) при $p_{r,i} = v_i \times 2^r$ выполняется с заложенным в сумму длин интервалов $H_{k,j}^*$ запасом

$$\delta|I_1| = (1 - \lambda)\delta(\sqrt{2} - 1) \times 2^a > 2^{a-s+1}/5$$

(см. (36)). При нормировке (делением не более, чем на 5×2^a) и подстановке u вместо v запас сокращается не более чем в 10×2^a раз, до величины не менее $\varepsilon = \frac{1}{25 \times 2^s}$.

Следовательно, при уменьшении свободных членов в приведенных неравенствах (38')–(40') на ε вектор u остается решением системы.

6. СОРТИРОВКА

Теорема 1. При любом $a \geq 2$ и $2^{a/4} \leq m \leq 2^{a/2}$ выполнено

$$P_m(a) \geq 2^{a+1/2} - O(a^{-\gamma} \times 2^a), \tag{41}$$

где γ мало, например, $\gamma = \frac{1}{5}$.

Доказательство. Воспользуемся результатом леммы 3 и применим лемму 2. Значения всех параметров заимствуются из леммы 3. Имеем $v \in T(\sigma[s])$ и $u \in T^\varepsilon(\sigma[s])$ для системы ограничений $\sigma[s]$ глубины $h = 2s + 3$ и ширины $w \leq s + 4$. При этом справедливо $u_i < 2^{y_i-1/2} < \pi(y_i)$ (см. (5)). Кроме того, $\varepsilon < 1 - \lambda < R = \|v - u\| < \sqrt{2}$. При $a < 100$ неравенство (41) заведомо выполнено при достаточно больших значениях константы под знаком “ O ”. Поэтому считаем, что $a \geq 100$. Положим $\varphi = \log\left(\frac{m\varepsilon}{4a^2}\right)$ и $s = \lceil 2\gamma \log a \rceil$. Если $\gamma \leq 1/4$, то при этом $m\varepsilon \geq 2^{a/4} / (50a^{2\gamma}) \geq 4a^2$, следовательно, $\varphi > 0$. Условие (16) леммы 2 выполнено автоматически.

Пусть $r + y_i \leq a - \frac{(a+2)\varphi \times 2^\varphi}{m} \leq r + y_i + \frac{1}{d}$, где $r \in \mathbb{N}$. С помощью леммы 2 получаем оценку

$$\begin{aligned} P_m(a) &\geq P_m\left[r + y_i + \frac{(r+2)\varphi \times 2^\varphi}{m}\right] \geq v_i \times 2^r - Rm \times 2^{\varphi+2} / \varepsilon - R \times 2^r e^{-\frac{\varepsilon}{2R}\left(\frac{\varphi}{w(d+1)(h+1)} - 2\right)} \geq \\ &\geq (1 - O(2^{-s/2})) \times 2^{a + \frac{1}{2} - \frac{(a+2)\varphi \times 2^\varphi}{m} - \frac{1}{d}} - O(m^2/a^2) - O\left(2^a e^{-\Theta(as^{-2} \times 2^{-2s})}\right) = \\ &= (1 - O(2^{-s/2})) \times 2^{a+1/2 - O(2^{-s})} - O(2^a/a^2) - 2^{a - O(as^{-2} \times 2^{-2s})} = 2^{a+1/2} (1 - O(a^{-\gamma})), \end{aligned}$$

если, скажем, $\gamma \leq 1/5$.

Следствие 1. При любом $a \geq 2$ и $2^{a/4} \leq m \leq 2^{a/2}$ выполнено

$$Q_{2m}(a) \geq 2^a - O(a^{-1/5} \times 2^a).$$

Реализованное доказательство главного технического результата (теорема 1) довольно громоздко. Поэтому для большей ясности приведем неформальное резюме схемы рассуждения.

1. Построим некоторую идеальную серию сравнений, всякий раз выполняя деление множества исходов пополам. К сожалению, она не дает сведения к подобной задаче меньшей размерности.

2. Отступление от идеала (поправочный коэффициент λ) позволяет разбалансировать длины интервалов, используя оценку (4) для сложности вставки, близкой к целому числу, и получить сходящийся рекурсивный алгоритм.

3. Дискретизация (параметр d) вводится, чтобы перейти от континуального к конечному семейству алгоритмов. В приведенном доказательстве шаги 2 и 3 совмещены. В результате мы имеем алгоритм с хорошей оценкой сложности, но который рекурсивно сводится к себе самому.

4. Обеспечим прогрессивный рост качества оценки сложности с увеличением размерности задачи, стартуя из точки, в которой оценка тривиально выполняется (лемма 2).

С технической стороны доказательство состоит в контролируемом учете поправок из трех источников: неравномерности разбиения, дискретизации и малой размерности.

Отметим, что даже упрощенный вариант стратегии из разд. 5 с выбором $s = 1$ и глубиной 6, только не универсальный, а с подбором оптимальных разбиений при каждом i , позволил бы улучшить результаты метода бинарных вставок при всех достаточно больших n . Однако для по-

лучения более точной оценки приходится выбирать параметр s растущим. Главный содержательный результат формулирует следующая

Теорема 2. При любых n выполняется равенство

$$S(n) = \log(n!) + O(n \log^{-1/5} n).$$

Доказательство. Начинаем как в методе бинарных вставок. Разобьем набор из n элементов на пары, упорядочим их и выполним сортировку старших элементов пар (они образуют главную цепочку). На это требуется $n/2 + S(n/2)$ сравнений. Пусть младшие элементы пар обозначаются через α_i с нумерацией в порядке возрастания ранга старших элементов в главной цепочке (фиг. 1).

Вставим первые $n_0 = n/\log n$ из младших элементов в главную цепочку методом бинарных вставок, потратив на это $n + O(n_0)$ сравнений. Оставшиеся элементы разобьем на группы по $m \approx \sqrt{n}/\log n$ штук: группа с номером k содержит элементы $\alpha_{n_0+(k-1)m+1}, \alpha_{n_0+(k-1)m+2}, \dots, \alpha_{n_0+km}$. Группы вставляются в главную цепочку по очереди в порядке возрастания номеров методом следствия 1. Последняя группа может содержать менее m элементов — их вставляем тривиальным бинарным методом.

Группа с номером t должна быть вставлена в интервал длины $L_t = 2(n_0 + tm) - m$. Согласно следствию 1, при некотором $\rho_t = \log L_t + O(\log^{-1/5} n)$ выполнено $Q_m(\rho_t) \geq L_t$. Поэтому сложность вставки t -й группы не превосходит $\rho_t m$.

Представим идеализированный случай, когда элемент α_k мог бы быть вставлен в главную цепочку за $\log(2k - 1)$ сравнений. Тогда, если вставлять элементы в порядке возрастания номеров, было бы затрачено $\log \prod_{k=1}^{\lceil n/2 \rceil} (2k - 1) = \log(n!) - \log((n/2)!) - n/2 + O(\log n)$ сравнений.

В групповом методе мы тратим на вставку элемента α_k не более $\log(2k + m) + O(\log^{-1/5} n) = \log(2k - 1) + O(\log^{-1/5} n)$ сравнений при $n_0 < k < n/2 - m$ и не более $\log(2k - 1) + O(1)$ сравнений при $k \leq n_0$ и $k \geq n/2 - m$. Общее превышение сложности вставки по отношению к идеализированной ситуации тогда можно оценить как

$$O(n_0 + m) + O((n/2 - n_0) \log^{-1/5} n).$$

Получаем рекуррентное соотношение

$$S(n) \leq n/2 + S(n/2) + \log(n!) - \log((n/2)!) - n/2 + O(n \log^{-1/5} n)$$

или, если выразить через величины $s(n) = S(n) - \log(n!)$,

$$s(n) \leq s(n/2) + O(n \log^{-1/5} n).$$

Отсюда следует $s(n) = O(n \log^{-1/5} n)$.

СПИСОК ЛИТЕРАТУРЫ

1. Ахо А., Хопкрофт Дж., Ульман Дж. Проектирование и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Кнут Д. Искусство программирования. Т. 3. Сортировка и поиск. М.: Вильямс, 2007.
3. Mehlhorn K. Data structures and algorithms. V. 1. Sorting and searching. Berlin, NY: Springer, 1984.
4. Ford L.R., Johnson S.M. A tournament problem // Amer. Math. Monthly. 1959. V. 66. № 5. P. 387–389.
5. PeczarSKI M. The Ford-Johnson algorithm still unbeaten for less than 47 elements // Inf. Process. Lett. 2007. V. 101. № 3. P. 126–128.
6. Schulte MönTing J. Merging of 4 or 5 elements with n elements // Theor. Comput. Sci. 1981. V. 14. P. 19–37.
7. Manacher G.K., Bui T.D., Mai T. Optimum combinations of sorting and merging // J. ACM. 1989. V. 36. № 2. P. 290–334.
8. Iwata K., Teruyama J. Improved average complexity for comparison-based sorting // Theor. Comput. Sci. 2020. V. 807. P. 201–219.
9. Edelkamp S., Weiß A., Wild S. QuickXsort: a fast sorting scheme in theory and practice // Algorithmica. 2020. V. 82. P. 509–588.
10. Graham R.L. On sorting by comparisons // in: Computers in Number Theory. London: Academic Press, 1971. P. 263–269.
11. Hwang F.K., Lin S. Optimal merging of 2 elements with n elements // Acta Inf. 1971. V. 1. P. 145–158.
12. Schönhage A., Paterson M., Pippenger N. Finding the median // J. Comp. Sys. Sci. 1976. V. 13. P. 184–199.
13. Dor D., Zwick U. Selecting the median // SIAM J. Comput. 1999. V. 28. № 5. P. 1722–1758.
14. Christen C. Improving the bounds for optimal merging // in: Proc. 19th IEEE Conf. on Found. of Comput. Sci. (Ann Arbor, USA, 16–18 October 1978). NY: IEEE, 1978. P. 259–266.