# Порядок средней сложности задачи интервального поиска на булевом кубе для одного алгоритма

Т.Д. Блайвас

Для задачи интервального поиска на булевом кубе предложен алгоритм построения решающей древовидной схемы по упорядоченной библиотеке над базисом переменных. Показано, что почти все построенные схемы имеют одинаковую по порядку временную сложность, меньшую сложности в классе сбалансированных древовидных схем [2].

## 1. Введение

В работе исследуется следующая задача информационного поиска. Имеется некоторое подмножество $V$ $n$-мерного булева куба $B_2^n$, называемое библиотекой. На булевом кубе берется произвольный интервал $(u, w)$, где $u = (u_1, \ldots, u_n)$, $w = (w_1, \ldots, w_n)$ и $u \preceq w$, то есть $u_i \leqslant w_i$ $i = 1, \ldots, n$. Требуется определить все такие элементы $y \in V$, $y = (y_1, \ldots, y_n)$, называемые записями, для которых выполнено $u \preceq y \preceq w$.

Приведем интерпретацию данной задачи. Допустим мы имеем частично разгаданный кроссворд, в котором все слова имеют одинаковую длину. Отгадываем слово, в котором известны не все буквы. Требуется найти в словаре такие слова, которые потенциально могут быть разгадываемым словом.

Задачу можно решать, если на каждом шаге алгоритма проверять условие $u_j \leqslant y_j \leqslant w_j$, $j \in \{1, \ldots, n\}$ для фиксированного набора компонент записи.

Предложен алгоритм построения древовидной решающей схемы, на вход которому поступает библиотека и порядок ее элементов. Показано, что для $k$-элементных библиотек ($k \leqslant 2^{n-2}$) из $n$-мерного ($n = \bar{o}(2^k)$) булева куба доля решающих деревьев, чья сложность по порядку равна $\left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}}$, асимтотически равна 1 при $n \to \infty$, $k \to \infty$.

## 2. Основные понятия и формулировка результатов

Мы будем использовать терминологию и обозначения из работы [3], но поскольку в данной работе рассматриваются только древовидные схемы, то здесь будет приведена несколько упрощенная версия понятия информационного графа.

Если $X$ — множество символов запросов с заданным на нем вероятностным пространством $\langle X, \sigma, \mathbf{P} \rangle$, где $\sigma$ — алгебра подмножеств множества $X$, $\mathbf{P}$ — вероятностная мера на $\sigma$; $Y$ — множество символов данных (записей); $\rho$ — бинарное отношение на $X \times Y$, называемое отношением поиска; то пятерка $S = \langle X, Y, \rho, \sigma, \mathbf{P} \rangle$ называется *типом*. Тройка $I = \langle X, V, \rho \rangle$, где $V$ — некоторое конечное подмножество множества $Y$, называемое библиотекой, называется задачей информационного поиска (ЗИП) типа $S$. Содержательно ЗИП $I = \langle X, V, \rho \rangle$ состоит в перечислении для произвольно взятого запроса $x \in X$ всех и точно тех записей $y \in V$, что $x \rho y$. Если $\mathcal{F}$ — суть множества символов одноместных предикатов, определенных на $X$, $\mathcal{F}$ называется базовым множеством и описывает множество элементарных операций, используемых при решении задачи информационного поиска.

Над базовым множеством $\mathcal{F}$ определяется понятие *информационного графа* (ИГ). В конечной многополюсной ориентированной сети выбирается вершина — полюс, называемая корнем. Остальные полюса называются листьями и им приписываются записи из $Y$. Ребрам ИГ приписываются предикаты из множества $\mathcal{F}$. Таким образом нагруженную многополюсную ориентированную сеть называют информационным графом над базовым множеством $\mathcal{F}$. Затем определяется *функционирование ИГ.* Предикатное ребро проводит запрос $x \in X$, если предикат ребра истинен на $x$; ориентированная цепочка ребер проводит $x$, если каждое ребро цепочки проводит $x$; запрос $x$ проходит в вершину $\beta$ ИГ, если существует ориентированная цепь, ведущая из корня в вершину $\beta$, которая проводит $x$; запись $y$, приписанная листу $\alpha$, попадает в ответ ИГ на $x$, если $x$ проходит в лист $\alpha$. Ответом ИГ $U$ на запрос $x$ называют множество записей, попавших в ответ $U$ на $x$, и обозначают его $\mathcal{J}_U(x)$. Эту функцию $\mathcal{J}_U(x)$ считают результатом функционирования ИГ $U$.

ИГ $U$ *разрешает* ЗИП $I = \langle X, V, \rho \rangle$, если $\mathcal{J}_U(x) = \{y \in V : x\rho y\}$.

Вводится *сложность ИГ.* Предикат $\varphi_\beta(x)$ истинный на $x$, если $x$ проходит в вершину $\beta$, и ложный в противном случае, называется функцией фильтра вершины $\beta$. *Сложностью ИГ $U$ на запросе $x \in X$* называется число $T(U, x) = \sum_{\beta \in \mathcal{R}} \psi_\beta \cdot \varphi_\beta(x)$, где $\mathcal{R}$ — множество вершин ИГ $U$, $\psi_\beta$ — количество ребер, исходящих из вершины $\beta$. Эта величина равна числу функций, вычисленных алгоритмом поиска, определяемым ИГ $U$, на запросе $x$.

Если каждая функция из $\mathcal{F}$ — измерима (относительно алгебры $\sigma$), то для любого ИГ $U$ над $\mathcal{F}$ функция $T(U, x)$ измерима.

*Сложностью ИГ $U$* называется математическое ожидание величины $T(U, x)$, равное $T(U) = \mathbf{M}_x\, T(U, x)$. Она характеризует среднее время поиска.

Легко показать, что

$$T(U) = \sum_{\beta \in U} \psi_\beta \cdot P(N_{\varphi_\beta}(x)). \tag{1}$$

Если $f$ предикат на множестве $X$, то $N_f(x) = \{x \in X : f(x) = 1\}$. Сложностью ребра, исходящего из вершины $\beta$ назовем число $P(N_{\varphi_\beta}(x))$. Согласно (1), сложность ИГ равна сумме сложностей ребер. Обозначим через $T_i(D)$, $i$ — натуральное число, сложность первых $i$ ярусов ребер дерева $D$.

Рассмотрим следующую ЗИП. Имеется некоторое $k$-элементное подмножество $n$-мерного булева куба $V \in B_2^n$ (библиотека). На булевом кубе задан некоторый интервал $(u, w)$, где $u = (u_1, \ldots, u_n)$, $w = (w_1, \ldots, w_n)$, и $u \preceq w$, то есть $u_i \leqslant w_i$, $\forall i = 1, \ldots, n$. Требуется определить все элементы $y \in V$, удовлетворяющие условию $u \preceq y \preceq w$.

Очевидно, что если $u_i = 1$ для некоторого $i$, то и $w_i = 1$, а, следовательно, и $y_i = 1$. Аналогично, если $w_i = 0$, то и $y_i = 0$. Таким образом, вышеописанная ЗИП сводится к следующей: есть библиотека $V \in B_2^n$, $|V| = k$, берем запрос $x = (x_1, \ldots, x_n)$ — трехзначный вектор, компоненты которого могут быть равны либо 1, либо 0, либо 2: если $u_i = 1$, то $x_i = 1$, если $w_i = 0$, то $x_i = 0$, иначе $x_i = 2$. Для данного запроса $x = (x_1, \ldots, x_n)$ хотим найти все $y = (y_1, \ldots, y_n) \in V$, для которых $y_i = x_i$, если $x_i = 1$ или $x_i = 0$, и $y_i$ любое из $\{0, 1\}$, если $x_i = 2$.

Получаем тип задач $S_n =< B_3^n, B_2^n, \rho, \sigma, P >$, где $B_3^n$ и $B_2^n$ трехзначный и двузначный (булев) кубы, соответственно, $\rho: \quad x\rho y \Leftrightarrow (x_i = y_i) \vee (x_i = 2)$, $\sigma$ — множество подмножеств $B_3^n$, и $P$ — равномерная вероятностная мера на $B_3^n$, то есть для любого $x \in B_3^n$ $P(x) = 3^{-n}$.

Вершину со степенью полуисхода, равной нулю назовем *висячей вершиной. Информационным деревом (ИД)* назовем ИГ без циклов, множество листьев которого совпадает с множеством висячих вершин, и все ребра которого ориентированы от корня к листьям. *Высотой ИД* назовем длину максимального пути из корня в лист.

Пусть $x \in \{0, 1, 2\}$ и $y \in \{0, 1, 2\}$. Определим функцию $x^y$ :

$$x^y = \begin{cases} x, & \text{если} \quad (y = 1) \& (x \neq 2) \\ \bar{x}, & \text{если} \quad (y = 0) \& (x \neq 2) \\ 1, & \text{если} \quad (y = 2) \vee (x = 2) \end{cases},$$

причем $\bar{x}$ понимается здесь как булевое отрицание.

Определим понятие яруса вершин. Вершиной первого яруса будем называть корень. Для любого числа $i$, не большего высоты дерева плюс 1, вершинами $i$-го яруса назовем такие вершины, из которых длина пути в корень равна $i-1$. Будем говорить, что ребро находится на ярусе с номером $i$, если оно исходит из вершины яруса с номером $i$.

Множество задач $I =< B_3^n, V, \rho >$ типа $S_n$, где $|V| = k$, обозначим через $\mathcal{I}(n, k)$.

Задачи из $\mathcal{I}(n,k)$ будем решать над базовым множеством $\mathcal{F}_0$ переменных: $\mathcal{F}_0 = \{x_1, \ldots, x_n, \bar{x}_1, ldots, \bar{x}_n\}$.

Пусть ребра информационного дерева занумерованы некоторым образом. *Номером цепи ребер* будем называть номером первого ребра цепи. *Нагрузкой цепи ребер* будем называть конъюнкцию нагрузок всех ребер цепи. *Полной цепью* из вершины $v$ будем называть цепь ребер, конечная вершина которой имеет полустепень исхода, не равную 1, а все остальные вершины цепи имеют полустепень исхода, в точности равную 1.

Будем рассматривать следующие функции, и пользоваться следующими обозначениями и операциями.

$E(v,N)$ — выдает полную цепь с номером $N$, исходящую из вершины $v$.
$V(v,N)$ — выдает конечную вершину полнуой цепи $E(v,N)$.
$N(v)$ — число ребер, исходящих из $v$.
$f(c) \in \mathcal{F}$ — нагрузка цепи $c$.
$X(c) = \{x_i^{s_i} | \quad x_i^{s_i}$ — множитель в $f(c)\}$.
$Y(i) = \{x_1^{y_1^i}, \ldots, x_n^{y_n^i}\}$.
$X'(c,i) = X(c) \bigcap Y(i)$.
$c(v_1, v_2)$ — цепь с начальной вершиной $v_1$ и конечной вершиной $v_2$.
Операция $C(c,i)$ удаляет полную цепь $c = (v_1, v_2)$, добавляет новые вершины $v'$ и $v''$, добавляет цепи $c_1 = (v_1, v')$, $c_2 = (v', v_2)$, $c' = (v', v'')$. Цепи $c_1$ соответствует нагрузка $\bigwedge_{x \in X'(c,i)} x$, цепи $c_2$ — $\bigwedge_{x \in X(e) \setminus X'(c,i)} x$, цепи $c'$ — $\bigwedge_{x \in Y(i) \setminus X'(c,i)} x$. Первое ребро цепи $c_1$ получает номер, равный номеру $c$, остальные ребра цепи $c$ получают номер 1; все ребра цепи $c_2$ получаюб номер, равный 1; первое ребро цепи $c'$ получает номер 2, все остальные ребра — номер 1.
Операция $NE(v,i,X)$ добавляет в вершину $v$ цепь с номером первого ребра $N(v)+1$, номерами остальных ребер, равными 1, и суммарной нагрузкой цепи $\bigwedge_{x \in X} x$. Листу $V(v, N(v)+1)$ приписываем запись $y_i$.

Для задачи $I =< B_2^n, V, \rho >\in \mathcal{I}(n,k)$ и перестановки элементов библиотеки $\sigma = (i_1, \ldots, i_k) \in S_k$, решающее информационное дерево над базисом $\mathcal{F}_0$ будем строить при помощи следующего алгоритма $A(V, \sigma)$.

**Алгоритм** $A(V, \sigma)$
Добавляем в множество вершин дерева корень $v_0$.
$NE(v_0, i_1, Y(i_1))$.
$p = 2$.
Шаг 0. $v = v_0$, $N = 1$, $Y = Y(i_p)$. Перейти к шагу 1.
Шаг 1. Если $X'(E(v, N), i_p) = \oslash$, то
Перейти к шагу 2.
Иначе
Перейти к шагу 3.
Шаг 2. Если $N < N(v)$
$N = N + 1$. Перейти к шагу 1.
Иначе
$NE(v, i_p, Y)$. Перейти к шагу 5.
Шаг 3. Если $X'(E(v, N), i_p) = X(E(v, N))$
$v = V(v, N)$, $N = 1$, $Y = Y \backslash X(E(v, N))$.
Перейти к шагу 1.
Иначе $C(E(v, N), i_p)$. Перейти к шагу 5.
Шаг 5. Если $p < k$
$p = p + 1$. Перейти к шагу 0.

Положим $\mathcal{V}(n, k) = \{V \in B_2^n, |V| = k\}$.

Пусть $V \in \mathcal{V}(n, k)$. Положим $\bar{T}(V) = M_\sigma T(A(V, \sigma))$, где $\sigma \in S_k$.

Обозначим через $\mathcal{A}(n, k)$ множество деревьев, получаемых алгоритмом $A$ для $k$-элементных библиотек из куба $B_2^n$ и всевозможных перестановок из $S_k$. Положим $\bar{T}(n, k) = M_V \bar{T}(V) = M_{D \in \mathcal{A}(n,k)} T(D)$, где $V \in B_2^n : |V| = k$.

**Теорема 1.** *При $n \to \infty$, $k \to \infty$, $n = \bar{o}(2^k)$, $n \geqslant 4$, $k \leqslant 2^{n-2}$*

$$\bar{T}(n, k) \asymp \left( \frac{k}{\log_2(kn)} \right)^{\log_2 \frac{4}{3}}.$$

## 3. Вспомогательные утверждения

Пусть $V'$ — упорядоченное множество. Через $A(V')$ обозначим результат работы алгоритма $A$ на неупорядоченной библиотеке $V$, состоящей из всех элементов $V'$, и перестановке $\sigma$, задающей порядок элементов в $V'$. *Бинарным деревом* будем называть древовидную схему, у которой из каждой вершины исходит не более двух ребер. Будем говорить, что базовая функция $x_i^{s_i}$ соответствует записи $y = (y_1, \dots, y_n)$, если $s_i = y_i$. Будем говорить, что запись $y = (y_1, \dots, y_n)$ удовлетворяет ребру $e$ с нагрузкой $x_i^{s^i}$, если $y_i = s_i$. Номером цепочки ребер назовем номер первого ребра цепочки.

**Лемма 1.** *Пусть $D \in \mathcal{A}(n,k)$. Тогда из каждой вершины дерева $D$ исходит не более двух ребер.*

**Доказательство.** Пусть $V = \{y^1, \dots, y^k\}$ — упорядоченная библиотека. Индукцией по построению дерева покажем, что из любой вершины выходит не более чем два ребра, а если $c_1 = (e_1^1, \dots, e_i^1)$ и $c_2 = (e_1^2, \dots, e_j^2)$ — полные цепочки ребер, выходящие из вершины $v$, и заканчивающеся вершинами с полустепенью исхода 0 или 2, с номерами 1 и 2 соответственно, то множество нагрузок второй цепочки содержит множество отрицаний нагрузок первой.

Базис индукции. Деревом для $A(\{y^1\}))$ будет являться цепочка ребер, следовательно, полустепени исхода любой вершины не более 1. При построении дерева $A(\{y^1, y^2\})$ мы получим цепь из корня (может быть нулевой длины), и из концевой вершины цепи — две цепочки ребер с противоположными нагрузками. Следовательно, утверждение леммы выполнено.

Индуктивный переход. Пусть построено дерево $A(\{y^1, \dots, y^{i-1}\})$, и для него выполнено условие индуктивного предположения. Строим дерево $A(\{y^1, \dots, y^i\})$. Если запись $y^i$ проходит в вершину $v$ дерева $A(\{y^1, \dots, y^{i-1}\})$, с полустепенью исхода 2, то если в цепочке с номером 1 не найдется ни одной базовой функции, подходящей для $y^i$, то она по индуктивному предположению найдется в цепочке с номером 2. Значит, полустепень исхода вершины не может увеличиться. Если запись удовлетворяет не всем ребрам цепочки $c$, то цепь разбивается некоторой вершиной $v$ на две цепи $c_1$ (входящая в $v$) и $c_2$ (исходящая из $v$), и из вершины $v$ тянется цепь $c'$, которая содержит все отрицания нагрузок цепи $c_2$, что соответствует индуктивному предположению.

Лемма 1 доказана.

Вершину $v$ ИД $D$ назовем *оптимально нагруженной*, если не существует такой базисной функции $x_i^{s_i}$, которая присутствовала бы в каждой цепочке поддерева, исходящего из $v$. Дерево $D$ назовем *оптимально нагруженным*, если любая его вершина с полустепенью исхода, равной 2, является оптимально нагруженной.

**Лемма 2.** *Все деревья из $\mathcal{A}(n, k)$ являются оптимально нагруженными.*

**Доказательство.** По утверждению леммы 1, любое дерево из $\mathcal{A}(n, k)$ является бинарным. Оптимальную нагруженность покажем индукцией по построению дерева. Пусть $V = (y^1, \ldots, y^k)$ — упорядоченная библиотека. Базис индукции. Пусть построено дерево $A(\{y^1\})$. Когда мы строим $A(\{y^1, y^2\})$, то мы ищем максимальное пересечение множеств значений $\{y_1^1, \ldots, y_n^1\}$ и $\{y_1^2, \ldots, y_n^2\}$, и этому пересечению будет соответствовать общая цепочка (может быть нулевой длины, если $y^1 = \bar{y}^2$), а две цепочки, исходящие из ее конца, вершины $v$, не будут иметь одинаковых функций в нагрузке. Следовательно, дерево $A(\{y^1, y^2\})$ является оптимально нагруженным.

Индуктивный переход. Пусть построено дерево $A(\{y^1, \ldots, y^{i-1}\})$, $i = 2, \ldots, k-1$. По предположению индукции оно является оптимально нагруженным. При построении $A(\{y^1, \ldots, y^i\})$, свойство оптимальной нагруженности вершин, из которых исходит два рера, не изменится, поскольку исходящие из нее цепочки уже не имеют нагрузочной функции, одинаковой для всех цепочек. Расммотрим единственную вершину $v$, которая в дереве $A(\{y^1, \ldots, y^{i-1}\})$ имела полустепень исхода 1, а в дереве $A(\{y^1, \ldots, y^i\})$ имеет полустепень исхода 2. По определению алгоритма, нагрузки, соответствующие записи $y^i$, и приписанные цепочке ребер, исходящей из вершины $v$, не будут пересекаться ни по какой функции с нагрузками цепочки, соответствующей поддереву, которое уже было в $A(\{y^1, \ldots, y^{i-1}\})$. Значит, для вершины $v$ выполняется свойство оптимальной нагруженности.

Лемма 2 доказана.

**Лемма 3.** *Для любого дерева $D \in \mathcal{A}(n, k)$ $T(D) \leqslant 10 \cdot k^{\log_2 \frac{4}{3}}$.*

**Доказательство.** Рассматриваем всевозможные информационные деревья над базисом переменных. Покажем, что если в дереве $D$ высоты $n$ с $k$ висячими вершинами существует цепочка не менее чем из 2 ребер, из конечной вершины которой исходит 2 ребра, то существует дерево $D'$ высоты $n$ с $k$ висячими вершинами, такое что $T(D) < T(D')$. Пусть в дереве $D$ цепь $c$ заканчивается ребром $e$ с концевой вершиной вершиной $v$, из которой исходят ребра $e_1$ и $e_2$. Пусть $v'$ — начальная вершина ребра $e$. Рассмотрим дерево $D'$, множество ребер которого получается из множества ребер $D$ удалением ребра $e$ и добавлением новых ребер $e'_1$ и $e'_2$, а множество вершин получается из множества вершин $D$ удалением вершины $v$ и добавлением вершин $v'_1$ и $v'_2$ таким образом, что вершина $v'_i$ инцедентна ребрам $e_i$ и $e'_i$, $i = 1, 2$, ребра $e'_1$ и $e'_2$ исходят из вершины $v'$ (рис 1).

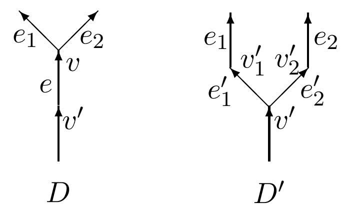


Рис. 1.

Пусть ребро $e$ находится на ярусе с номером $i$ в дереве $D$. Получаем, что в дереве $D'$ на ярусе с номером $i$ число ребер увеличилось на 1, на остальных ярусах число ребер в деревьях $D$ и $D'$ совпадают. Из (1) следует, что сложность дерева равна сумме сложностей его ребер. Вероятность попасть в вершину яруса с номером $i$ равна $\frac{2^{i-1} \cdot 3^{n-i+1}}{3^n} = \left(\frac{2}{3}\right)^{i-1}$, так как в вершину яруса $i$ проходят те запросы, которые для каждой из $i-1$ нагрузок цепочки ребер либо удовлетворяют этой нагрузке, либо равны 2, остальные $n-i+1$ координат запроса могут принимать любое значение из $\{0, 1, 2\}$; количество всех запросов равно $3^n$. Следовательно, $T(D') - T(D) = \left(\frac{2}{3}\right)^{i-1} > 0$. Получаем, что из всех усеченных бинарных деревьев высоты $n$ с $k$ висячими вершинами наибольшую сложность имеют деревья, у которых полустепени всех вершин на ярусах с номерами не большими $[\log_2 k]$ равны 2, на ярусе с номером $]\log_2 k[$ полустепени исхода не больше двух, а на ярусах с номерами, большими $]\log_2 k[$ — равны 1. Следовательно, если $D \in \mathcal{A}(n,k)$, то

$$T(D) \leqslant \sum_{i=1}^{]\log_2 k[} \left(\frac{2}{3}\right)^{i-1} + 2^{]\log_2 k[} \cdot \sum_{i=]\log_2 k[+1}^{n} \left(\frac{2}{3}\right)^{i-1} =$$
$$= \frac{3}{2} \sum_{i=1}^{]\log_2 k[-1} \left(\frac{4}{3}\right)^{i-1} + 3\left(\frac{4}{3}\right)^{]\log_2 k[} \left(\frac{2}{3}\right)^{n-[\log_2 k]} \leqslant$$
$$\leqslant \frac{9}{2}\left(\frac{4}{3}\right)^{]\log_2 k[} + 3\left(\frac{4}{3}\right)^{]\log_2 k[} \leqslant$$
$$\leqslant \frac{15}{2}\left(\frac{4}{3}\right)^{\log_2 k+1} = 10 \cdot k^{\log_2 \frac{4}{3}}.$$

Лемма 3 доказана.

Обозначим через $T_i'(D)$ сумму сложностей ярусов ребер с номерами от $i$ до $n$ в дереве $D$.

**Лемма 4.** *При* $n \to \infty$, $k \to \infty$, $i \geqslant \log_2 k + \log_2 \log_2(kn)$ *выполнено*

$$T_i'(D) = \bar{o}\left(\left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}}\right).$$

**Доказательство.** Число ребер на каждом ярусе не превышает $k$. Вероятность пройти в вершину яруса с номером $i$ равна $\left(\frac{2}{3}\right)^{i-1}$. Поэтому

$$T_i'(D) \leqslant k\left(\frac{2}{3}\right)^{[\log_2 k+\log_2\log_2 k]} \sum_{j=0}^{n-[\log_2 k+\log_2\log_2 k]} \left(\frac{2}{3}\right)^{j} \leqslant$$
$$\leqslant \frac{3}{2}k\left(\frac{2}{3}\right)^{\log_2 k+\log_2\log_2 k} = \frac{3}{2}\cdot k^{\log_2\frac{4}{3}}(\log_2(kn))^{\log_2\frac{2}{3}} =$$
$$= \frac{3}{2}\cdot\frac{k^{\log_2\frac{4}{3}}}{(\log_2(kn))^{\log_2\frac{3}{2}}} = \bar{o}\left(\left(\frac{k}{\log_2(kn)}\right)^{\log_2\frac{4}{3}}\right).$$

Лемма 4 доказана.

Обозначим через $\chi^p_{k,l,k'}$ долю разбиений числа $k$ на $l$ слагаемых, $p$ из которых в точности равны 1, а остальные $l-p$ не меньше $k'$.

**Лемма 5.** *При* $n \to \infty$, $k \to \infty$, $2\log_2 \log_2 k \leqslant k' \leqslant \log_2 k$, $l \leqslant \frac{k}{\log_2 k}$,

*при* $l =< \sqrt{\frac{k}{(\log_2 k)^\alpha}}$, $\alpha > 1$ *и* $p > 0$,
*при* $\sqrt{\frac{k}{(\log_2 k)^\alpha}} \leqslant l \leqslant c\sqrt{k}$, $\alpha > 1$, $c \in \mathbb{R}$ *и* $p > \log_2 k$,
*при* $k = \bar{o}(l^2)$ *и* $p > 2\frac{l^2}{k}$ *выполнено*

$$\chi^p_{k,l,k'} = \bar{o}\left(\frac{1}{\log_2 k}\right).$$

**Доказательство.**

$$\chi^p_{k,l,k'} = \frac{C_l^p \cdot C^{l-p-1}_{k-p-(l-p)k'-1}}{C^{l-1}_{k-1}} =$$

$$= \frac{l!(k-p-(l-p)k'-1)!(k-l)!(l-1)!}{p!(l-p)!(k-l-(l-p)k'-1)!(l-p-1)!(k-1)!} =$$

$$= \frac{(l-p+1)\cdots l(k-l-(l-p)k'+1)\cdots(k-p-(l-p)k'-1)(l-p)\cdots(l-1)}{p!\cdot(k-l+1)\cdots(k-p+1)\cdot(k-p)\cdots(k-1)} =$$

$$\leqslant \frac{1}{p!}\left(\frac{k-p-(l-p)k'-1}{k-p-1}\right)^{l-p-1}\left(\frac{l^2}{k}\right)^p =$$

$$= \frac{1}{p!}\left(1-\frac{(l-p)k'}{k-p-1}\right)^{l-p-1}\left(\frac{l^2}{k}\right)^p.$$

Поскольку $k' = \bar{o}(\frac{k}{l})$, то $(l-p)k' = \bar{o}(k) = \bar{o}(k-p)$, и

$$\left(1-\frac{(l-p)k'}{k-p-1}\right)^{l-p-1} \sim \frac{1}{e^{(l-p)k'\frac{l-p-1}{k-p-1}}} \sim \frac{1}{e^{\frac{(l-p)^2k'}{k-p-1}}} \sim \frac{1}{e^{\frac{(l-p)^2k'}{k}}}.$$

Получаем, что при $k \to \infty$

$$\chi^p_{k,l,k'} < \frac{1}{p!}\cdot\frac{1}{e^{\frac{(l-p)^2k'}{k}}}\cdot\left(\frac{l^2}{k}\right)^p.$$

Рассмотрим три случая.

1) $l < \sqrt{\frac{k}{(\log_2 k)^\alpha}}$, $\alpha > 1$. Тогда $\frac{l^2}{k} < \frac{1}{(\log_2 k)^\alpha}$. И если $p \geqslant 1$, то

$$\chi^p_{k,l,k'} \leqslant \frac{1}{\log_2^\alpha k} = \bar{o}\left(\frac{1}{\log_2 k}\right).$$

2) $\sqrt{\frac{k}{(\log_2)^\alpha}} < l < \sqrt{ck}$, $c \in \mathbb{R}$. Если $p > \log_2 l$, то

$$\chi_{k,l,k'} < \frac{1}{(\log_2 l)!} c^{\log_2 l} = \frac{c \cdot c \cdots c}{1 \cdot 2 \cdots \log_2 l} =$$
$$= \frac{c^2}{1 \cdot \log_2 l} \cdot \frac{c^2}{2(\log_2 l - 1)} \cdots \frac{c^2}{\log_2^2 (l-1)} <$$
$$< \left(\frac{c^2}{\log_2 l}\right)^{\frac{l}{2}} < \left(\frac{2c^2}{\log_2 k - \alpha \log_2 \log_2 k}\right)^{\frac{l}{2}} = \bar{o}\left(\frac{1}{\log_2 k}\right).$$

3) $k = \bar{o}(l^2)$. Если $p > 2\frac{l^2}{k}$, то $p$ — растущая функция от $k$ и $p! \sim \frac{p^p}{\sqrt{2\pi p}}$.

$$\chi^p_{k,l,k'} < \sqrt{2\pi p} \cdot \frac{1}{e^{k'}} \cdot \frac{1}{2^p} \leqslant \frac{1}{e^{\log_2 \log_2 k}} = \frac{1}{(\log_2 k)^{\log_2 e}} = \bar{o}\left(\frac{1}{\log_2 k}\right).$$

Лемма 5 доказана.

**Лемма 6.**

$$\max_{k_1 + \cdots + k_p = k} \prod \left(1 - \frac{1}{2^{k_i}}\right) = \left(1 - \frac{1}{2^{\frac{k}{p}}}\right)^p.$$

**Доказательство.** Покажем, что функция действительных переменных $x$ и $y$ $f(x,y) = \left(1 - \frac{1}{2^x}\right)\left(1 - \frac{1}{2^y}\right)$ при ограниении $x + y = k$, принимает наибольшее значение, если $x = y$.

$$f(x,y) = \left(1 - \frac{1}{2^x}\right)\left(1 - \frac{1}{2^y}\right) =$$
$$= \left(1 - \frac{1}{2^x}\right)\left(1 - \frac{1}{2^{k-x}}\right) = 1 - \frac{1}{2^x} - \frac{1}{2^{k-x}} + \frac{1}{2^k} =$$
$$= 1 + \frac{1}{2^k} - \frac{2^x + 2^{k-x}}{2^k} = g(x).$$

Функция $g(x)$ имеет максимум при том же значении $x$, при котором минимальна сумма $2^x + 2^{k-x}$.

$$\left(2^x + 2^{k-x}\right)' = 0 \Leftrightarrow \ln 2 \cdot 2^x - \ln 2 \cdot 2^{k-x} = 0 \Leftrightarrow 2^{2x} = 2^k,$$

причем производная отрицательна при $x < \frac{k}{2}$, и положительна при $x > \frac{k}{2}$. Следовательно, максимальное значение $f(x,y)$ принимается в точке $\left(\frac{k}{2}, \frac{k}{2}\right)$. Рассмотрим функцию от $p$ переменных $f(k_1, \dots, k_p) = \prod \left(1 - \frac{1}{2^{k_i}}\right)$. Пусть верно $k_1 + \dots + k_p = k$. Допустим, при заданном ограничении на переменные, $f(k_1, \dots, k_p)$ принимает максимальное значение на наборе $a_1, \dots, a_p$, и существуют $i \neq j$, такие что $a_i \neq a_j$. Возьмем $a'_i = a'_j = \frac{a_i + a_j}{2}$. Тогда $a'_i + a'_j$, но

$$\left(1 - \frac{1}{2^{a_i}}\right)\left(1 - \frac{1}{2^{a_j}}\right) < \left(1 - \frac{1}{2^{a'_i}}\right)\left(1 - \frac{1}{2^{a'_j}}\right),$$

что противоречит максимальности функции $f$.

Лемма 6 доказана.

**Лемма 7.** *Пусть в усеченном бинарном дереве $D$, решающем задачу $I \in \mathcal{I}(n,k)$, на каждом ярусе вершин с номером большим $i$, два ребра выходят менее чем из $\frac{1}{3}$ вершин. Тогда $T(D) < 10 \cdot T_i(D)$.*

**Доказательство.** Обозначим через $E_j$ число ребер на ярусе с номером $j$ в дереве $D$. Вероятность попасть в вершину яруса с номером $i+1$ равна $\frac{2^i \cdot 3^{n-i}}{3^n} = \left(\frac{2}{3}\right)^i$, так как в такую вершину проходят только те из $3^n$ записей, у которых $i$ координат определены нагрузками предыдщих $i$ ярусов, и могут принимать значение, соответствующее нагрузке, или «2»; остальные $n-i$ координат принимают любое значение из $\{0,1,2\}$. Следовательно, учитывая (1), суммарная сложность всех ребер яруса с номером $(i+1)$ равна $E_{i+1}\left(\frac{2}{3}\right)^i$, $i = 0, \cdots, n-1$. Из условий теоремы получаем, что $E_{j+1} < \frac{4}{3} \cdot E_j$ при $j > i$. Тогда

$$T_i(D) < T(D) = \sum_{j=1}^{n} \left(\frac{2}{3}\right)^{j-1} E_j \leqslant$$

$$\leqslant \sum_{j=1}^{i} \left(\frac{2}{3}\right)^{j-1} E_j + \sum_{j=i+1}^{n} \left(\frac{2}{3}\right)^{j-1} \left(\frac{4}{3}\right)^{j-i} E_i =$$

$$= T_i(D) + \left(\frac{2}{3}\right)^{i-1} E_i \sum_{j=i+1}^{n} \left(\frac{8}{9}\right)^{j-i} <$$

$$< T_i(D) + 9\left(\frac{2}{3}\right)^{i-1} E_i < 10 \cdot T_i(D).$$

Лемма 7 доказана.

Обозначим через $\mu(n,k)$ долю $k$-элементных подмножеств булева куба размерности $n$, которые целиком содержатся в хотя бы в одном подкубе размерности, меньшей $n$.

**Лемма 8.** *Пусть* $n = \bar{o}(2^k)$, $n \geqslant 4$. *Тогда при* $k \leqslant 2^{n-2}$

$$\frac{2n}{5^k} < \mu(n,k) \leqslant \frac{2n}{2^k}.$$

**Доказательство.** Для того, чтобы некоторая библиотека $V$ принадлежала подкубу размерности, меньшей $n$, необходимо, чтобы все ее элементы совпадали в одной из $n$ координат. Для фиксированного значения координаты, можно выбрать $C_{2^{n-1}}^k$ подмножеств размера $k$ в подкубе размерности $n-1$. Поскольку каждая координата может принимать точно два значения, доля библиотек, содержащихся в подкубе, заданном какой-либо координатой, равна $2\frac{C_{2^{n-1}}^k}{C_{2^n}^k}$. Пусть $k \leqslant 2^{n-2}$. По формуле включения-исключения,

$$\mu(n,k) = 2\frac{C_n^1 C_{2^{n-1}}^k}{C_{2^n}^k} + \cdots + (-1)^{]n-\log_2 k[} 2^{[n-\log_2 k]} \frac{C_n^{[n-\log_2 k]} C_{2^{]\log_2 k[}}^k}{C_{2^n}^k}.$$

Следовательно, выполняются следующие неравенства.

$$2\frac{C_n^1 C_{2^{n-1}}^k}{C_{2^n}^k} - 4\frac{C_n^2 C_{2^{n-2}}^k}{C_{2^n}^k} < \mu(n,k) \leqslant 2\frac{C_n^1 C_{2^{n-1}}^k}{C_{2^n}^k},$$

$$2n\frac{(2^{n-1}-k+1)\cdots 2^{n-1}}{(2^n-k+1)\cdots 2^n}-4\frac{n^2}{2}\frac{(2^{n-2}-k+1)\cdots 2^{n-2}}{(2^n-k+1)\cdots 2^n}<$$
$$<\mu(n,k)\leqslant 2n\frac{(2^{n-1}-k+1)\cdots 2^{n-1}}{(2^n-k+1)\cdots 2^n},$$

$$2n\left(\frac{2^{n-1}-k+1}{2^n-k+1}\right)^k-2n^2\left(\frac{2^{n-2}}{2^n}\right)^k<\mu(n,k)\leqslant 2n\left(\frac{2^{n-1}}{2^n}\right)^k,$$
$$2n\left(\frac{2^{n-1}-2^{n-2}-1}{2^n-1}\right)^k-\bar{o}\left(\frac{n}{2^k}\right)<\mu(n,k)\leqslant\frac{2n}{2^k}$$

при $n\to\infty$, $k\to\infty$. Так как

$$\frac{2^{n-2}-1}{2^n-1}=1-\frac{2^n-2^{n-2}}{2^n-1}=1-\frac{3\cdot 2^{n-2}}{2^n-1}=1-\frac{3}{4-2^{2-n}}\geqslant\frac{1}{5}$$

при $n\geqslant 4$, то

$$\frac{2n}{5^k}<\mu(n,k)\leqslant\frac{2n}{2^k}$$

при $k\to\infty$.

Лемма 8 доказана.

Обозначим через $E_i^{(V,\sigma)}$ число ребер на ярусе с номером $i$ в дереве $A(V,\sigma)$, $i=1,\ldots,n$. Тогда $T(A(V,\sigma))=\sum_{i=1}^n\left(\frac{2}{3}\right)^{i-1}E_i^{(V,\sigma)}$. Положим

$$E_i(k,n)=M_{(V,\sigma)}E_i^{(V,\sigma)}=\frac{\sum_{V\in\mathcal{V}(n,k)}\sum_{\sigma\in S_k}E_i^{(V,\sigma)}}{k!C_{2^n}^k}.$$

**Лемма 9.** *При* $i\leqslant[\log_2 k-\log_2\log_2(kn)]$ *верно* $E_i(k,n)\sim 2^i$.

**Доказательство.** Пусть $\mathcal{D}_i(n,k)$ — множество деревьев из $\mathcal{A}(n,k)$, содержащих на ярусе с номером $i$ асимтотически $2^i$ ребер. Обозначим $D_i(n,k)=\frac{|\mathcal{D}_i(n,k)|}{k!C_{2^n}^k}$.

Для того, чтобы библиотека в каждом решающем дереве из $\mathcal{A}(n,k)$, имела 2 ребра на первом ярусе, необходимо, чтобы она не содержалась ни в одном подкубе, размерности, меньшей $n$. В силу леммы 8

$$|\mathcal{D}_1(n,k)| > \left(1 - \frac{2n}{2^k}\right) C_{2^n}^k \cdot k! \sim C_{2^n}^k \cdot k!.$$

Следовательно,

$$D_1(n,k) > 1 - \frac{2n}{2^k} \sim 1.$$

Значит,

$$E_1(n,k) = \frac{\sum_{V\in\mathcal{V}(n,k)} \sum_{\sigma\in S_k} E_1^{(V,\sigma)}}{k! C_{2^n}^k} \geqslant \left(1 - \frac{2n}{2^k}\right) \cdot 2 \sim 2.$$

Предположим, что выполнено

$$D_{i-1}(n,k) > \left(1 - \bar{o}\left(\frac{1}{\log_2 k}\right)\right)^{i-1} \left(1 - \frac{2n}{2^{k'}}\right)^{2^{i-1}}.$$

Разобьем все деревья из $\mathcal{D}_{i-1}(n,k)$ на классы эквивалентности следующим образом. Два дерева попадут в один класс, если их корневые поддеревья высоты $i-1$ полностью совпадают, в том числе совпадают нагрузки и нумерация ребер. Таким образом, в каждом классе находятся деревья с одинаковым началом. Берем произвольный класс. Пусть $l \sim 2^{i-1}$ — количество концевых ребер. Из деревьев класса возьмем только те, у которых число вершин, в которые проходит не менее $k' = \frac{\log_2(kn)}{\log_2\log_2\log_2 k}$ записей асимтотически равно $l \sim 2^{i-1}$ при $i \leqslant \log_2 k - \log_2\log_2(kn)$. По утверждению леммы 5, при $i \leqslant \log_2 k - \log_2\log_2(kn)$ доля таких деревьев составляет не менее чем $1 - \bar{o}\left(\frac{1}{\log_2 k}\right)$, так как

$$k' \cdot 2^i < \frac{\log_2(kn)}{\log_2\log_2\log_2 k} \cdot \frac{k}{\log_2(kn)} = \frac{k}{\log_2\log_2\log_2 k} = \bar{o}(k).$$

Для каждой вершины, в которую проходит более чем $\frac{\log_2(kn)}{\log_2\log_2\log_2 k}$ записей, вероятность того, что в вершине не реализуется подкуб размерности меньшей, чем $n-i$, не менее чем

$$1 - \mu(n-i, \frac{\log_2(kn)}{\log_2\log_2\log_2 k}) \geqslant 1 - \frac{2(n-i)}{\frac{kn}{\log_2\log_2 k}} \geqslant 1 - \frac{2\log_2(kn)}{k}.$$

Следовательно, в каждом классе, доля деревьев из $\mathcal{D}_i(n,k)$ не меньше чем
$$\left(1-\bar{o}\left(\frac{1}{\log_2 k}\right)\right)\left(1-\frac{2\log_2\log_2 k}{k}\right)^{2^{i-1}}.$$

Значит и для всего множества $\mathcal{D}_i(n,k)$ выполнено

$$D_i(n,k) > D_{i-1}(n,k)\left(1-\bar{o}\left(\frac{1}{\log_2 k}\right)\right)\left(1-\frac{2\log_2\log_2 k}{k}\right)^{2^{i-1}} >$$
$$> \left(1-\bar{o}\left(\frac{1}{\log_2 k}\right)\right)^i\left(1-\frac{2\log_2\log_2 k}{k}\right)^{2^{i}} >$$
$$> \left(1-\bar{o}\left(\frac{1}{\log_2 k}\right)\right)^i\left(1-\frac{2\log_2\log_2 k}{k}\right)^{\frac{k}{\log_2(kn)}} \sim 1$$

при $i \leqslant \log_2 k - \log_2\log_2(kn)$. Значит, доля деревьев из $D_b(n,k)$, для которых усеченное бинарное решающее дерево имеет асимптотически $2^i$ ребер на ярусе с номером $i \leqslant \log_2 k - \log_2\log_2(kn)$, не меньше чем $D_i(n,k) \sim_k 1$. Тогда

$$E_i = \frac{\sum_{V\in\mathcal{V}(n,k)}\sum_\sigma E_i^{(V,\sigma)}}{k!C_{2n}^k} \geqslant$$
$$\geqslant \left(1-\bar{o}\left(\frac{1}{\log_2 k}\right)\right)^i\left(1-\frac{2\log_2\log_2 k}{k}\right)^{\frac{k}{\log_2(kn)}} 2^i \sim 2^i.$$

Лемма 9 доказана.

Обозначим через $\mathcal{D}'(n,k)$ подмножество деревьев из $\mathcal{A}(n,k)$, таких, что если $D \in \mathcal{D}'(n,k)$, то $T(D) \leqslant 66\frac{k}{\log_2(kn)}$.

**Лемма 10.** *При $n\to\infty$, $k\to\infty$*

$$|\mathcal{D}'(n,k)| \succeq 1 - \frac{1}{exp(k^\varepsilon)}$$

*для некоторого* $0<\varepsilon<\frac{\log_2 5}{3}$.

**Доказательство.** Положим $i_{\max} = ]\log_2 k - \log_2\log_2(kn) + \log_2 9[$. Возьмем все деревья из $\mathcal{A}(n,k)$, у которых на ярусе с номером $i_{\max}$ асимтотически $2^{i_{\max}} \geqslant \frac{9k}{\log_2(kn)}$ ребер. Все деревья разобьем на классы эквивалентности. Два дерева попадут в один класс, если корневые поддеревья высоты $i_{\max}$ одинаковы, в том числе одинаковы нагрузки и нумерация ребер, и если количество записей, проходящих в одинаковые вершины, является равным. Для каждого класса эквивалентности рассмотрим $p \geqslant \frac{3k}{\log_2(nk)}$ вершин на ярусе $i_{\max}$, в которые проходит максимальное количество — $k_{j_1},\ldots,k_{j_p}$ — записей. Пусть в некоторую вершину $v$, из которой исходит дерево высоты $n' = n - i_{\max}$, проходит $k'$ записей. Если все записи содержатся хотя бы в одном подкубе, размерности меньшей $n'$, то по лемме 2 у любого дерева, выходящего из вершины $v$ и построенного для данных записей алгоритмом $A$, из корня будет выходить точно одно ребро. Значит, доля деревьев, имеющих точно одно исходящее ребро из вершины $v$, равна доле библиотек, принадлежащих некоторому подкубу, размерности, меньшей чем $n'$. Тогда доля деревьев из класса, у которых из всех данных вершин выйдет два ребра, по лемме 8 и лемме 6 не больше чем

$$\prod_{i=1}^{p}\left(1-\frac{2(n-i_{\max})}{5^{k_i}}\right) < \left(1-\frac{2(n-i_{\max})}{5^{\frac{k}{p}}}\right)^p \sim \frac{1}{e^{\frac{p}{5^{\frac{k}{p}}}}}$$

В свою очередь,

$$\frac{k}{p} < \frac{\log_2(kn)}{3},\quad 5^{\frac{k}{p}} < (kn)^{\frac{\log_2 5}{3}},\quad \frac{p}{5^{\frac{k}{p}}} > \frac{k^{1-\frac{\log_2 5}{3}}}{\log_2(kn)} > k^{\varepsilon} \to \infty$$

для некоторого $0 < \varepsilon < 1 - \frac{\log_2 5}{3} \approx 0,226$ при $k \to \infty$. Следовательно,

$$\frac{1}{exp(\frac{p}{5^{k/p}})} \to 0$$

при $k \to \infty$. Значит, у более чем $1 - \frac{1}{exp(k^{\varepsilon})}$ деревьев из класса доля вершин, из которых выйдет два ребра, не превосходит $\frac{3k}{\log_2 k}$, то есть трети числа всех ребер. Очевидно, что если $D$ — произвольное дерево из $\mathcal{A}(n,k)$, имеющее асимтотически $2^{i_{\max}}$ на ярусе с номером $i_{\max}$, то для любого яруса с номером $i \geqslant i_{\max}$ количество ребер не менее $\frac{9k}{\log_2(kn)}$, и с вероятностью более чем $1 - \frac{1}{exp(k^{\varepsilon})}$, менее трети всех вершин яруса $i$ имеют полустепень исхода, равную 2. Поскольку, в соответствии с утверждением леммы 4, для любого дерева сложность последних $n - \log_2 k - \log_2 \log_2(kn)$ ярусов ребер равна $\bar{o}\left(\left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}}\right)$, то доля деревьев, удовлетворяющих условиям леммы 7, не менее

$$\left(1 - \frac{1}{exp(k^{\varepsilon})}\right)^{2\log_2 \log_2(kn)} \sim 1 - \frac{2\log_2 \log_2(kn)}{exp(k^{\varepsilon})}.$$

По лемме 7 сложность дерева $D$ по порядку равна сложности первых $]\log_2 k - \log_2 \log_2(kn) + \log_2 9[$ ярусов. То есть

$$T_d(I) \leqslant 10 \cdot 2 \cdot \sum_{i=1}^{]\log_2 k - \log_2 \log_2(kn) + \log_2 9[} \left(\frac{4}{3}\right)^{i-1} =$$

$$\leqslant 20 \cdot \left(\frac{4}{3}\right)^{\log_2 9 + 1} \left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}} < 66 \left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}}.$$

Заметим, что оценка числа деревьев с нужной нам сложностью не зависит от выбора класса. Следовательно, доля всех деревьев со сложностью не более $66\left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}}$ при $n \to \infty$, $k \to \infty$, составляет не менее $1 - \frac{2\log_2 \log_2(kn)}{exp(k^{\varepsilon})}$, $\varepsilon > 0$.

Лемма 10 доказана.

## 4. Доказательство теоремы

Поскольку $|\{V \in \mathcal{V}(n,k)\}| = C_{2^n}^k$, то

$$\bar{T}(n,k) = \frac{\sum_{V \in \mathcal{V}(n,k)} \bar{T}(V)}{C_{2^n}^k} = \frac{\sum_{V \in \mathcal{V}(n,k)} \sum_{\sigma \in S_k} T(A(V,\sigma))}{k! C_{2^n}^k}.$$

Положим $i_{\max} = [\log_2 k - \log_2 \log_2(kn)]$. Из утверждения леммы 9 следует, что $E_i(n,k) \sim 2^i$ при $i \leqslant i_{\max}$. Тогда при $n \to \infty$, $k \to \infty$

$$\bar{T}(n,k) = \frac{\sum_{V \in \mathcal{V}(n,k)} \sum_{\sigma \in S_k} \sum_{i=1}^{n} \left(\frac{2}{3}\right)^{i-1} E_i^{(V,\sigma)}}{k! C_{2^n}^k} =$$

$$= \frac{\sum_{i=1}^{n} \left(\frac{2}{3}\right)^{i-1} \cdot \left(\sum_{V \in \mathcal{V}(n,k)} \sum_{\sigma \in S_k} E_i^{(V,\sigma)}\right)}{k! C_{2^n}^k} =$$

$$= \sum_{i=1}^{n} \left(\frac{2}{3}\right)^{i-1} E_i \geqslant \sum_{i=1}^{i_{\max}} \left(\frac{2}{3}\right)^{i-1} E_i \sim \sum_{i=1}^{i_{\max}} \left(\frac{2}{3}\right)^{i-1} 2^i = \frac{3}{2} \sum_{i=1}^{i_{\max}} \left(\frac{4}{3}\right)^{i-1} \geqslant$$

$$\geqslant \frac{3}{2} \cdot 3 \left(\frac{4}{3}\right)^{\frac{k}{\log_2(kn)} - 1} = \frac{27}{8} \left(\frac{k}{\log_2(kn)}\right)^{\log_2 \frac{4}{3}}.$$

С другой стороны, из утверждений леммы 10 и леммы 3 следует, что при $n \to \infty$, $k \to \infty$

$$\bar{T}(n,k) \leqslant \left(1 - \frac{2 \log_2 \log_2(kn)}{exp\,(k^{\varepsilon})}\right) \cdot 66 \left(\frac{k}{\log_2(nk)}\right)^{\log_2 \frac{4}{3}} +$$

$$+ \frac{10 \cdot k^{\log_2 \frac{4}{3}} \cdot 2 \cdot \log_2 \log_2(kn)}{exp(k^{\varepsilon})} \sim 66 \left(\frac{k}{\log_2(nk)}\right)^{\log_2 \frac{4}{3}}.$$

Значит,

$$\bar{T}(n,k) \asymp \left(\frac{k}{\log_2(nk)}\right)^{\log_2 \frac{4}{3}}.$$

Теорема доказана.

В заключение автор выражает благодарность Э.Э. Гасанову за постановку задачи и научное руководство.

## Список литературы

[1] Блайвас Т.Д. Решение задачи интервального поиска на булевом кубе // Проблемы теоретической кибернетики. Тезисы докладов XIII Международной конференции. Часть I. С. 22.

[2] Блайвас Т.Д. Оптимальное решение задачи интервального поиска на булевом кубе в классе сбалансированных древовидных схем // Интеллектуальные системы. Т. 7. Вып. 1–4. 2002–2003. С. 223–244

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