Лекция 5.

Задача поиска идентичных объектов.
Алгоритм бинарного поиска.
Константный в среднем алгоритм
поиска идентичных объектов.

1 Поиск идентичных объектов

В данном разделе мы будем рассматривать задачу поиска идентичных объектов, которая в том или ином виде встречается во всех информационных системах и базах данных. Задача поиска идентичных объектов состоит в поиске в информационном массиве объекта, идентичного объекту-запросу.

Как уже говорилось, для задачи поиска идентичных объектов (как и для задачи о близости) в рамках АДВ-модели справедлива логарифмическая теоретико-информационная нижняя оценка сложности для худшего случая. Поэтому считается, что бинарный поиск является оптимальным по порядку для задачи поиска идентичных объектов, и это как бы закрывает проблему, но несмотря на это имеется много работ, посвященных исследованию алгоритмов поиска идентичных объектов в худшем случае. Связано это, во-первых, с проблемой поддержки сбалансированности бинарного дерева при операциях вставки и удаления, а во-вторых, с тем, что бинарный поиск хорош только тогда, когда целиком вся библиотека помещается во внутренней памяти (*внутренний поиск*), если же библиотека вся или частично расположена на внешних носителях (*внешний поиск*), то эффективность бинарного поиска сразу падает. Также много работ, посвященных изучению алгоритмов поиска идентичных объектов, имеющих хорошие временные характеристики в среднем. Связаны они в основном с методом хеширования, на котором мы остановимся чуть подробнее.

Хеширование предполагает наличие хеш-функции. Хеш-функция $h(y)$ определена на множестве записей X и переводит его в множество $\{\overline{1,m}\}$, где m — параметр хеш-функции.

Обычно используется два основных метода хеширования. Первый — предложенный А. Думи и называемый методом цепочек, предполагает наличие m списков. Тогда для поступившего запроса x (он же запись) вычисляется значение хеш-функции $h(x)$, и если оно равно i, то просматривается i-ый список и в нем ищется запись x. Если это поиск с занесением, то в случае неудачного поиска запись x добавляется к i-ому списку.

Второй метод хеширования предложен А. П. Ершовым и называется открытой адресацией. Он предполагает наличие закольцованного массива для m записей и наличие понятия пустой записи. После этого для поступившего запроса x вычисляется значение хеш-функции $h(x)$, и если оно равно i, то просматривается с i-ой позиции массив записей пока запись x будет найдена или пока не встретится пустая запись. Если это поиск с занесением, то в случае неудачного поиска на место встреченной пустой записи помещается запись x.

Методы хеширования хороши тем, что при удачном выборе хеш-функции, равномерно рассеивающей поступающие записи, время поиска в среднем будет очень малым. Вместе с тем Д. Кнут усматривает три основных недостатка метода хеширования.

а) После неудачного поиска мы знаем лишь то, что нужной записи нет, тогда как с помощью бинарного поиска мы обнаруживаем ближайших соседей ненайденной записи, что часто бывает важно во многих приложениях.

б) Часто довольно трудно распределить память под хеш-таблицу. Если выделить слишком мало, то она может переполниться, и потребуется тягостное "рехеширование". Если выделить слишком много, то это расточительно.

в) "Наконец, при использовании методов хеширования нужно свято верить в теорию вероятностей, ибо они эффективны лишь в среднем, а худший случай просто ужасен!— цитата из Д. Кнута. Поэтому они не всегда подходят для работы в реальном масштабе времени, например, для управления движением транспорта, поскольку на карту поставлены человеческие жизни. Алгоритмы, использующие сбалансированные деревья, гораздо безопаснее, ведь они имеют гарантированную верхнюю границу времени поиска.

В данной работе предлагается метод поиска идентичных объектов, который в среднем эффективен, как хорошие методы хеширования, то есть обеспечивает мгновенное решение, а в худшем случае такой же, как метод бинарного поиска, и плюс к этому не обладает недостатком а), то есть в случае неудачного поиска выходит к ближайшим соседям ненайденной записи, что позволяет использовать этот алгоритм для решения задач о близости.

Опишем формально задачу поиска идентичных объектов.

Пусть нам дано множество X, на котором задано отношение линейного порядка $\preceq$, то есть такое бинарное отношение на $X \times X$, которое для любых $x, y, z \in X$ удовлетворяет условиям

- рефлексивности $x \preceq x$;
- транзитивности $(x \preceq y)\&(y \preceq z) \to (x \preceq z)$;
- антисимметричности $(x \preceq y)\&(y \preceq x) \to (x = y)$;
- связности $(x \preceq y) \vee (y \preceq x)$.

Рассмотрим следующий тип задач поиска $S_{id} = \langle X, X, \rho_{id} \rangle$, где отношение поиска ρ_{id} есть отношение идентичности, то есть

$$x \rho_{id} y \Longleftrightarrow x = y.$$

Тип S_{id} будем называть *типом поиска идентичных объектов.*

2 Бинарный поиск

Если V — конечное подмножество X, то через $\max\limits_{y \in V} y$ будем обозначать такое $y_{max} \in V$, что для любых $y \in V$ $y \preceq y_{max}$, и через $\min\limits_{y \in V} y$ обозначим такое $y_{min} \in V$, что для любых $y \in V$ $y_{min} \preceq y$.

Пусть

$$g_{\preceq,a}(x) = \begin{cases} 1, & \text{если } x \preceq a \\ 2 & \text{в противном случае} \end{cases}, a \in X, \tag{1}$$

$$f_{=,a}(x) = \begin{cases} 0, & \text{если } x \neq a \\ 1, & \text{если } x = a \end{cases}, a \in X. \tag{2}$$

$$G_1 = \{g_{\preceq,a}(x) : a \in X\}, \tag{3}$$

$$F = \{f_{=,a}(x) : a \in X\}, \tag{4}$$

$$\mathcal{F}_{bin} = \langle F, G_1 \rangle. \tag{5}$$

Справедлива следующая теорема.

Теорема 1. *Если $I = \langle X, V, \rho_{id} \rangle$ — задача поиска идентичных объектов, то есть задача типа S_{id}, $\mathcal{F}_{bin}$ — базовое множество, определяемое соотношениями (1)–(5), то*

$$T(I, \mathcal{F}_{bin}, 2|V| - 1) \leq] \log_2 |V| [+ 1,$$

$$\widehat{T}(I, \mathcal{F}_{bin}, 2|V| - 1) \leq] \log_2 |V| [+ 1.$$

Доказательство: Пусть $|V| = k$.

Построим ИГ $D^1(V)$, имеющий вид дерева, следующим образом.

Возьмем бинарное сбалансированное дерево с k концевыми вершинами, высоты $] \log_2 k[$, все ребра которого ориентированы от корня к концевым вершинам. Если в этом дереве есть внутренняя вершина, из которой исходят ребра, ведущие одно в концевую вершину, а другое во внутреннюю (если такая вершина есть, то она только одна), то из концевой вершины, в которую ведет одно из этих ребер, выпустим одно ребро. Полученное дерево обозначим D. Объявим концевые вершины этого дерева листьями и сопоставим им слева направо в порядке возрастания элементы из V. (Говоря о дереве, мы подразумеваем некоторую его укладку, и направления "влево", "вправо"определяются уже на этой укладке.)

Пусть β — произвольная внутренняя вершина дерева D. Обозначим через V_β множество записей, соответствующих листьям, схемно достижимым из β. Пусть β' — вершина, в которую ведет левое (если оно одно, то единственное) ребро из β. Пусть

$$y_\beta = \max_{y \in V_{\beta'}} y.$$

Объявим все внутренние вершины дерева D, из которых исходят ребра, ведущие во внутренние вершины, вершинами переключения и для каждой такой вершины β левому ребру, из нее выходящему, припишем 1, а правому — 2, а самой вершине припишем переключатель $g_{\preceq, y_\beta}(x)$.

Все ребра дерева D, входящие в листья, объявим предикатными и припишем каждому такому ребру, ведущему в лист с записью y, предикат $f_{=,y}(x)$.

ИГ, полученный из дерева D после определения таким образом нагрузки, обозначим $D^1(V)$ и будем называть первым деревом бинарного поиска.

Покажем, что $D^1(V)$ решает задачу $I = \langle X, V, \rho_{id} \rangle$.

Возьмем произвольную запись $y \in V$. Пусть α — лист $D^1(V)$, которому приписана запись y. В этот лист ведет единственная цепь, состоящая из не более чем $]\log_2 k[$ ребер, причем все эти ребра, кроме последнего, переключательные. Последнее ребро в цепи — предикатное с предикатом $f_{=,y}(x)$, и его проводимость на запросе y равна $f_{=,y}(y) = 1$. Покажем, что проводимость остальных ребер цепи на запросе y равна 1. Возьмем произвольно одно из этих ребер (β, β'). Если (β, β') — левое, исходящее из β , то $y \in V_{\beta'}$ и предикат, приписанный вершине β, $g_{\preceq, y_\beta}(y) = 1$, так как $y \preceq y_\beta = \max\limits_{y' \in V_{\beta'}} y'$. Если (β, β') — правое ребро, исходящее из β, то $y \not\preceq y_\beta$ и $g_{\preceq, y_\beta}(y) = 2$, тем самым проводимость ребра (β, β') в обоих случаях равна 1. Следовательно, $\varphi_\alpha(y) = 1$. Так как ребро, ведущее в лист α, имеет нагрузку $f_{=,y}$, то для любого запроса $x \neq y$ $\varphi_\alpha(x) = 0$.

Таким образом, мы показали, что $\varphi_\alpha(x) = f_{=,y}(x)$. Учитывая произвольность листа α и критерий допустимости информационных графов, получим, что ИГ $D^1(V)$ решает задачу $I = \langle X, V, \rho_{id} \rangle$.

Подсчитаем объем ИГ $D^1(V)$.

Поскольку $D^1(V)$ есть дерево, в котором полустепень исхода любой вершины равна 2, кроме, быть может, одной вершины, то если в дереве $D^1(V)$ нет вершины с полустепенью исхода 1, то в $D^1(V)$ будет ровно $2 \cdot k - 2$ ребра, а если в дереве $D^1(V)$ есть вершина с полустепенью исхода 1, то в $D^1(V)$ будет $2 \cdot k - 1$ ребро. Отсюда следует, что

$$Q(D^1(V)) \leq 2k - 1. \tag{6}$$

Подсчитаем сложность ИГ $D^1(V)$.

Рассмотрим произвольный запрос $x \in X$. Как мы показали выше, активные на этом запросе вершины графа $D^1(V)$ (вершина β активна на запросе, если $\varphi_\beta(x) = 1$) образуют единственную цепь. Причем в этой цепи не более чем $]\log_2 k[-1$ переключательных вершин и одна вершина, из которой исходит не более двух ребер с предикатами. Тем самым

$$T(D^1(V), x) \leq]\log_2 k[+1.$$

Отсюда с учетом (6) сразу следует, что

$$T(I, \mathcal{F}_{bin}, 2k - 1) \leq]\log_2 k[+1,$$

$$\widehat{T}(I, \mathcal{F}_{bin}, 2k-1) \leq] \log_2 k[+1.$$

Тем самым теорема доказана.

Отметим, что первое дерево бинарного поиска соответствует стандартному алгоритму бинарного поиска в версии Боттенбрука.

3 Константный в среднем алгоритм поиска

Пусть $\mathbf{N}_0 = \mathbf{N} \cup \{0\}$ — множество целых неотрицательных чисел.

Пусть

$$L_1(l) = \begin{cases} 0, & \text{если } l = 0 \\ ] \log_2 l[+1, & \text{если } l = 1, 2, 3 \\ \log_2 l + 2, & \text{если } l \geq 4 \end{cases} - \quad (7)$$

функция, определенная на множестве $\mathbf{N}_0$.

Докажем следующее вспомогательное утверждение.

Лемма 1. *Пусть $L_1(l)$ — функция, определенная выше, пусть $k, m \in \mathbf{N}$, пусть*

$$r_1(k,m) \stackrel{\text{def}}{=} \max\{\sum_{i=1}^{m} L_1(l_i) : l_1 \in \mathbf{N}_0, \ldots, l_m \in \mathbf{N}_0, \sum_{i=1}^{m} l_i = k\}.$$

Тогда

$$\begin{aligned} r_1(k,m) &= \left(k - \left[\frac{k}{m}\right] \cdot m\right) \cdot L_1\left(\left[\frac{k}{m}\right] + 1\right) + \\ &\quad + \left(m - k + \left[\frac{k}{m}\right] \cdot m\right) \cdot L_1\left(\left[\frac{k}{m}\right]\right). \end{aligned}$$

Доказательство: Если доопределить функцию $L_1(l)$ на положительную полуось числовой прямой, например, следующим образом:

$$L_1(x) = \begin{cases} x, & \text{если } 0 \leq x \leq 4 \\ \log_2 x + 2, & \text{если } x \geq 4 \end{cases},$$

то полученная функция будет непрерывной и вогнутой. И, вообще говоря, вогнутость этой функции объясняет результат леммы.

Более подробное доказательство будем вести от противного.

Предположим, что лемма неверна, то есть

$$r_1(k,m) = \sum_{i=1}^{m} L_1(l'_i) > \left(k - \left[\frac{k}{m}\right]\cdot m\right)\cdot L_1\left(\left[\frac{k}{m}\right]+1\right) + \\ + \left(m - k + \left[\frac{k}{m}\right]\cdot m\right)\cdot L_1\left(\left[\frac{k}{m}\right]\right), \tag{8}$$

и среди чисел $l'_i (i = \overline{1,m})$ существует 2 числа, разность которых не меньше двух.

Без ограничения общности можем считать, что $l'_1 - l'_2 \geq 2$.

Пусть

$$l''_1 = \left]\frac{l'_1 + l'_2}{2}\right[,\ l''_2 = \left[\frac{l'_1 + l'_2}{2}\right].$$

$l''_1 + l''_2 = l'_1 + l'_2$ и $l''_1 - l''_2 \leq 1$.

Так как функция $L_1(x)$ вогнутая, то по свойству вогнутых функций

$$L_1(l''_1) + L_1(l''_2) \geq L_1(l'_1) + L_1(l'_2). \tag{9}$$

Если в неравенстве (9) неравенство строгое, то получили противоречие, так как $\sum_{i=1}^{m} L_1(l'_i)$ — не максимально; если же нет, то обозначим $l''_i = l'_i$ $(i = \overline{3,m})$ и перейдем к рассмотрению суммы

$$\sum_{i=1}^{m} L_1(l''_i) = \sum_{i=1}^{m} L_1(l'_i) = r_1(k,m).$$

Если среди чисел l''_i $(i = \overline{1,m})$ нет пары чисел, разность которых превышает 1, то получим противоречие с неравенством (8), если же есть, то опять выполним операцию, описанную выше, и избавимся от такой пары, и так будем делать до тех пор, пока либо не получим строгое неравенство в неравенстве (9), либо не придем к разбиению $l_1^{(n)}, \ldots, l_m^{(n)}$, такому, что

$$\sum_{i=1}^{m} L_1(l_i^{(n)}) = r_1(k,m),$$

и все они отличаются не более чем на 1, и тем самым получим противоречие с неравенством (8), что доказывает лемму 1.

Пусть на X задано вероятностное пространство $\langle X, \sigma, \mathbf{P}\rangle$.

Пусть $g_m^1(x)$ — переключатель такой, что

$$g_m^1(x) = i, \text{ если } x \in X_i \ (i = \overline{1,m}), \tag{10}$$

где $X_1, \ldots, X_m$ — разбиение множества X (то есть $X = X_1 \cup \cdots \cup X_m$ и $X_i \cap X_j = \emptyset$, если $i \neq j$) такое, что

$$\mathbf{P}(X_i) \leq c/m \ (i = \overline{1,m}), \tag{11}$$

где c — постоянная, не зависящая от m.

Так как

$$\mathbf{P}(X) = \sum_{i=1}^{m} \mathbf{P}(X_i) \leq c,$$

то отсюда сразу следует, что $c \geq 1$.

Пусть

$$G_2 = \{g_m^1(x) : m \in \mathbf{N}\}, \tag{12}$$

$$\mathcal{F} = \langle F, G_1 \cup G_2\rangle. \tag{13}$$

Справедлива следующая теорема.

Теорема 2. *Пусть $I = \langle X, V, \rho_{id}\rangle$ — задача поиска идентичных объектов, то есть задача типа S_{id}, где $|V| = k$, $\mathcal{F}$ — базовое множество, определяемое соотношениями (1)–(4), (10)–(13), m — натуральное число, c — константа, определяемая соотношением (11), $L_1(l)$ — функция, определяемая соотношением (7). Тогда*

$$\begin{aligned} 1 \ &\leq \ T(I, \mathcal{F}, 2 \cdot k + m - 1) \leq \\ &\leq \ \frac{c}{m}\left(\left(k - \left[\frac{k}{m}\right] \cdot m\right) \cdot L_1\left(\left[\frac{k}{m}\right] + 1\right) + \right. \\ &\quad \left. + \left(m - k + \left[\frac{k}{m}\right] \cdot m\right) \cdot L_1\left(\left[\frac{k}{m}\right]\right)\right) + 1. \end{aligned}$$

В частности,

$$1 \leq T(I, \mathcal{F}, (2 + c) \cdot k) \leq 2$$

и $T(I, \mathcal{F}) \sim 1$ при $k \to \infty$. Кроме того, для ИГ $U \in \mathcal{U}(I, \mathcal{F})$, на котором достигается верхняя оценка, $\widehat{T}(U) \leq 2+] \log_2 k[$.

Доказательство: Построим ИГ U_m^0, имеющий вид дерева, следующим образом.

Возьмем вершину β_0 и объявим ее корнем графа U_m^0. Выпустим из β_0 m ребер, припишем им числа от 1 до m, объявим β_0 точкой переключения и припишем ей переключатель $g_m^1(x)$.

Пусть $V_i = X_i \cap V,\ l_i = |V_i|,\ i = \overline{1,m}$.

Конец ребра с номером i обозначим β_i.

Для всех таких i, что $V_i \neq \emptyset$, выпустим из вершины β_i первое дерево бинарного поиска, описанное в разделе 2, то есть дерево $D^1(V_i)$.

Полученный ИГ с $|V| = k$ листьями обозначим U_m^0. Это граф над базовым множеством $\mathcal{F}$.

Покажем, что U_m^0 решает задачу $I = \langle X, V, \rho_{id} \rangle$.

Возьмем произвольную запись $y \in V$. Пусть $y \in V_i$ ($i \in \{\overline{1,m}\}$), то есть лист α, которому приписана запись y, принадлежит дереву $D^1(V_i)$. Так как $g_m^1(y) = i$, запрос y пройдет по i-му ребру, исходящему из корня, в корень дерева $D^1(V_i)$. Так как $D^1(V_i)$ решает задачу поиска идентичных объектов для библиотеки V_i, то только запрос y пройдет в лист α.

Таким образом, мы показали, что $\varphi_\alpha(x) = f_{=,y}(x)$. Учитывая произвольность выбора y и критерий допустимости информационных графов, получим, что ИГ U_m^0 решает задачу $I = \langle X, V, \rho_{id} \rangle$.

Подсчитаем объем графа U_m^0.

$$Q(U_m^0) \leq m + \sum_{i=1}^{m} \max(0, 2 \cdot l_i - 1) \leq 2 \cdot k + m - 1.$$

Подсчитаем сложность ИГ U_m^0.

Рассмотрим произвольный запрос $x \in X$. Пусть $x \in X_i$ ($i = \overline{1,m}$).

$$T(U_0^m, x) = 1 + T(D^1(V_i), x) \leq 2 +] \log_2 l_i [\leq 1 + L_1(l_i), \qquad (14)$$

где $L_1(l)$ — функция, определяемая соотношением (7).

По определению и учитывая лемму 1,

$$\begin{aligned}
T(U_0^m) &= \mathbf{M}_x\, T(U_0^m, x) = \int_X T(U_0^m, x)\, \mathbf{P}(dx) = \\
&= \sum_{i=1}^{m} \int_{X_i} T(U_0^m, x)\, \mathbf{P}(dx) \le \sum_{i=1}^{m} (1 + L_1(l_i)) \cdot \mathbf{P}(X_i) = \\
&= 1 + \sum_{i=1}^{m} L_1(l_i) \cdot \mathbf{P}(X_i) \le 1 + \frac{c}{m} \sum_{i=1}^{m} L_1(l_i) \le \\
&\le 1 + r_1(k,m)\, \frac{c}{m} = \\
&= \frac{c}{m} \left(\left(k - \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] + 1 \right) + \right. \\
&\quad \left. + \left(m - k + \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] \right) \right) + 1.
\end{aligned}$$

Возьмем $m =]c \cdot k[$. При этом

$$Q(U_0^m) \le 2k +]c \cdot k[- 1 \le (2 + c) \cdot k.$$

Так как $c \ge 1$, то $m \ge k$. Если $m = k$, то $k/m = 1$ и

$$r_1(m,k) = 0 \cdot L_1(2) + k \cdot L_1(1) = k \cdot L_1(1).$$

Если $m > k$, то $k/m = 0$ и

$$r_1(m,k) = k \cdot L_1(1) + (m - k) \cdot L_1(0) = k \cdot L_1(1).$$

Следовательно,

$$T(I, \mathcal{F}, (2 + c) \cdot k) \le T(U_0^m) \le 1 + \frac{c}{]c\,k[}\, k \cdot L_1(1) \le 2.$$

Если взять $m =]k \cdot \gamma(k)[$, где $\gamma(k) \to \infty$ при $k \to \infty$ и $\gamma(k) > 1$ при любом k, то

$$T(I, \mathcal{F}) \le 1 + \frac{c \cdot k}{]k \cdot \gamma(k)[} \lesssim 1.$$

С другой стороны, $T(I, \mathcal{F}) \ge 1$, так как из корня должно исходить хотя бы одно ребро, поскольку $k > 0$.

И наконец, оценим B-сложность ИГ U_0^m, воспользовавшись соотношение (14),

$$\widehat{T}(U_0^m) = \max_{x \in X} T(U_0^m, x) \leq 2 + \max_{1 \leq i \leq m}] \log_2 l_i [\leq 2 +] \log_2 k [.$$

Тем самым теорема доказана.

Следствие 1. *Если $\mathcal{F}$ — базовое множество, определяемое соотношениями (1)–(4), (10)–(13), k — натуральное число, то*

$$1 \leq T(k, S_{id}, \mathcal{F}) \leq \mathcal{T}(k, S_{id}, \mathcal{F}) \leq 2$$

и при $k \to \infty$

$$T(k, S_{id}, \mathcal{F}) \sim \mathcal{T}(k, S_{id}, \mathcal{F}) \sim 1.$$

Приведем два примера задачи поиска идентичных объектов.

Пример 1. Пусть $S_{id1} = \langle X, X, \rho_{id} \rangle$ — тип поиска идентичных объектов, где $X = [0, 1]$, причем задано вероятностное пространство $\langle X, \sigma, \mathbf{P} \rangle$ над X такое, что $\mathbf{P}$ задается функцией плотностью вероятности $p(x)$.

Пусть

$$\begin{aligned} X_1 &= \{x \in X : 0 \leq x \leq 1/m\}. \\ X_i &= \{x \in X : \frac{i-1}{m} < x \leq \frac{i}{m}\},\ i = \overline{2, m}. \end{aligned} \tag{15}$$

Тогда переключатель $g_m^1(x) = \max(1,]x \cdot m[)$ удовлетворяет соотношению (10).

Если $p(x) \leq c = const$, то $\mathbf{P}(X_i) \leq c/m$, и мы находимся в условиях теоремы 2.

Приведем неформальное описание алгоритма поиска, описанного в доказательстве теоремы 2, применительно к задаче $I = \langle X, V, \rho_{id} \rangle$ типа S_{id1}, где $|V| = k$.

Разобьем отрезок $[0, 1]$ на m равных частей в соответствии с соотношениями (15).

Каждой части сопоставим подмножество множества V, состоящее из точек, принадлежащих этой части.

Теперь для произвольной точки-запроса x поиск точки из V, идентичной запросу, будем вести следующим образом.

Определим ту часть отрезка, которой принадлежит запрос x. Ее номер равен $\max(1,]x \cdot m[)$.

Далее во множестве, сопоставленном найденной части, осуществим обычный бинарный поиск.

Согласно лемме 1 наибольшее среднее время поиска будет в том случае, когда точки из множества V равномерно распределены по всем m частям отрезка.

Если в качестве m взять $m = k$, то случай, когда в каждую часть попадет по одной точке из V, будет иметь наибольшую сложность. А поскольку найти или не найти объект во множестве мощности 1 можно за 1 шаг, то в среднем за 2 шага (на первом мы определили номер нужной части) мы можем решить задачу поиска идентичных объектов.

Отметим, что в случае неудачного поиска алгоритм выводит к ближайшим соседям ненайденной точки.

Если внимательно присмотреться к описанному алгоритму, то можно заметить, что по своей структуре он напоминает хеширование по методу цепочек, в котором в качестве хеш-функции взято умножение, а цепочки-списки организованы в виде бинарного дерева.

Пример 2. Пусть $S_{id2} = \langle X, X, \rho_{id} \rangle$ — тип поиска идентичных объектов, где $X = \{1, \ldots, N\}$.

Пусть $\langle X, \sigma, \mathbf{P} \rangle$ — вероятностное пространство над X, где σ — множество всех подмножеств, а вероятностная мера $\mathbf{P}$ определяет равномерную вероятность на множестве запросов X, то есть для любого $x \in X$ $\mathbf{P}(x) = 1/N$.

Пусть задано $m \in \mathbf{N}$.

Пусть $r = N - m \cdot [N/m]$.

Пусть $X_1, \ldots, X_m$ — такое разбиение множества, что

$$\begin{aligned} X_i &= \{x \in X : 1 + (i-1) \cdot ([N/m] + 1) \le x \le i \cdot ([N/m] + 1)\}, \\ &\quad i = \overline{1, r}, \\ X_i &= \{x \in X : r \cdot ([N/m] + 1) + 1 + (i - 1 - r) \cdot [N/m] \le x \le \\ &\quad \le r \cdot ([N/m] + 1) + (i - r) \cdot [N/m]\},\ i = \overline{r+1, m}, \end{aligned}$$

и $g_m^1(x) = i$, если $x \in X_i,\ i \in \{\overline{1, m}\}$. Тогда

$$\mathbf{P}(X_i) \le ([N/m] + 1)/N < 2/m,$$

и мы опять находимся в условиях теоремы 2 при $c = 2$.

Упражнения

1. Пусть $I = \langle X, V, \rho_c \rangle$ — задача о близости, где $X = \{1, 2, ..., N\}$, $V \subseteq X$, ρ_c — отношение поиска, задаваемое соотношением на $X \times V$ и определяемое соотношением

$$x \rho_c y \Longleftrightarrow (y \in V) \& (x \leq y) \& (\neg(\exists y')((y' \in V) \& (x \leq y') \& (y' < y))), \qquad (16)$$

т.е. $x \rho_c y$, если $y \in V$, ближайшее справа к x. По аналогии с поиском идентичных объектов постройте ИГ, решающий задачу о близости I методом бинарного поиска. Получите соответствующие оценки сложности.

2. Пусть $I = \langle X, V, \rho_c \rangle$ — задача о близости, где $X = \{1, 2, ..., N\}$, $V \subseteq X$, ρ_c — отношение поиска, задаваемое соотношением на $X \times V$ и определяемое соотношением (16). По аналогии с поиском идентичных объектов постройте ИГ, решающий задачу о близости I и имеющий константную сложность.