Лекция 6.

Константный в худшем случае
алгоритм поиска идентичных объектов.
Оценки памяти константного в худшем случае
алгоритма поиска идентичных объектов.

1 Константный в худшем случае алгоритм поиска

Теорема 1. *Пусть*

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

$$F = \{f_{=,a}(x) : a \in X\}, \tag{2}$$

$$g_m^1(x) = i, \text{ если } x \in X_i \ (i = \overline{1,m}), \tag{3}$$

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 $i \neq j$),

$$G_2 = \{g_m^1(x) : m \in \mathbf{N}\}. \tag{4}$$

Если $I = \langle X, V, \rho_{id} \rangle$ — задача поиска идентичных объектов, $\mathcal{F} = \langle F, G_2 \rangle$ — базовое множество, определяемое соотношениями (1),(2),(3),(4), такое, что существует такое натуральное число m, что для любых различных $y, y' \in V$ $g_m^1(y) \neq g_m^1(y')$, то $1 \leq \widehat{T}(I, \mathcal{F}) \leq 2$.

Доказательство: Рассмотрим ИГ U, построенный следующим образом. Возьмем вершину, которую объявим корнем ИГ. Пусть m — число, упомянутое в условиях теоремы. Выпустим из корня m ребер, объявим их переключательными и занумеруем подряд, начиная с 1. Припишем

корню переключатель g_m^1. Для каждой записи $y \in V$ из конца ребра с номером $g_m^1(y)$ выпустим предикатное ребро с предикатом $f_{=,y}$, конец этого ребра объявим листом и припишем этому листу запись y. Понятно, что U решает ЗИП I, и для любого запроса $x \in X$ выполняется $T(U,x) \le 2$. Следовательно, $\widehat{T}(I,\mathcal{F}) \le 2$. Но поскольку одно вычисление всегда надо сделать, то $\widehat{T}(I,\mathcal{F}) \ge 1$.

Тем самым теорема доказана.

Следствие 1. *Если $\mathcal{F} = \langle F, G_2 \rangle$ — базовое множество, определяемое соотношениями (1),(2),(3),(4), такое, что для любой библиотеки V существует такое натуральное число m, что для любых различных записей $y, y' \in V$ $g_m^1(y) \ne g_m^1(y')$, то для любого натурального k выполнено*

$$1 \le \widehat{T}(k, S_{id}, \mathcal{F}) \le \widehat{T}(k, S_{id}, \mathcal{F}) \le 2.$$

2 Оценки памяти константного в худшем случае алгоритма поиска

В данном разделе рассматривается задача поиска идентичных объектов в ее геометрической интерпретации, которая звучит следующим образом. Дано конечное подмножество $V = \{y_1, \ldots, y_k\}$ точек из отрезка $[0,1]$ вещественной прямой. Требуется построить условный алгоритм, который для произвольной точки $x \in [0,1)$ (эта точка называется запросом) позволяет найти номер точки из множества V, которая совпадает с x (если такая точка в V существует), при условии, что мы умеем выполнять следующие операции над вещественными числами: арифметические операции (сложение, вычитание, умножение, деление, взятие целой части вещественного числа), операции сравнения и возможно некоторые другие простейшие операции. При этом допускается предобработка данных, которая может состоять в сортировке данных (множества V), а также в построении некоторых дополнительных структур.

В данном разделе детализируется алгоритм, описанный в предыдущем пункте, применительно к данной геометрической интерпретации, и показывается, что для почти всех задач поиска идентичных объектов (то есть при вариации множества V) данный алгоритм позволяет при объеме памяти порядка k^2 решать задачу в худшем случае за константное число элементарных операций, то есть операций, которые обычно используются в компьютерах. Здесь под объемом памяти понимается количество

ячеек для хранения вещественных чисел, куда можно поместить данные и дополнительные структуры, а худший случай берется по множеству всех возможных значений запроса, то есть по множеству $[0, 1)$. Здесь для упрощения дальнейшего изложения мы выбросили точку 1 из множества запросов.

Опишем предлагаемый алгоритм. Пусть нам дано множество $V = \{y_1, y_2, \ldots, y_k\}$, в котором производится поиск. Это множество будем называть библиотекой. Выполним следующую предобработку. Упорядочим точки из V в порядке возрастания и, чтобы не усложнять обозначения, далее считаем, что $y_1 < y_2 < \cdots < y_k$. Находим число $d_V = \min\limits_{2 \le i \le k} (y_i - y_{i-1})$. Пусть $m =]1/d_V[$ — наименьшее целое, не меньшее, чем $1/d_V$. Выделим место под массив целых длины m, и элементы этого массива будем обозначать n_i $(i = 0, 1, 2, \ldots, m-1)$. Разделим отрезок $[0, 1]$ на m равных частей:

$$A_i = [i/m, (i+1)/m),\ i = 0, 1, \ldots, m-2,\ A_{m-1} = [(m-1)/m, 1].$$

В каждую часть может попасть не более одной точки из множества V. Теперь заполним массив n_i следующим образом:

$$n_i = \begin{cases} -1, & \text{если в } A_i \text{ не попало ни одной точки из } V \\ q & \text{в противном случае, где } q \text{ — номер точки} \\ & \text{из } V, \text{ которая попала в } A_i, \end{cases}$$

где $i = 0, 1, 2, \ldots, m-1$.

После того как сделана данная предобработка, поиск будем осуществлять следующим образом. Пусть нам дан запрос $x \in [0, 1)$. Вычислим $j = [x \cdot m]$ — целая часть числа $x \cdot m$. Понятно, что $x \in A_j$. Если n_j равно -1, то в библиотеке V нет числа равного x. В противном случае сравниваем y_{n_j} с x и если они равны, то номер n_j является ответом задачи, иначе ответ пуст. Тем самым в худшем случае мы выполняем одну операцию умножения, одну операцию взятия целой части, одну операцию сравнения целых чисел, одну операцию сравнения вещественных чисел и две операции извлечения элемента массива, всего 6 элементарных операций. Объем памяти, необходимый данному алгоритму, равен сумме объемов массивов целых чисел длины m и вещественных чисел длины k. Ниже приводятся результаты, оценивающие величину m.

Пусть k — натуральное число, большее 1 и $\xi_1, \xi_2, \ldots, \xi_k$ — независимые

равномерно распределенные на отрезке $[0,1]$ случайные величины. Пусть

$$d(\xi_1,\ldots,\xi_k)=\min_{1\le i<j\le k}|\xi_i-\xi_j|.$$

Пусть r — вещественное число и $f(k,r)=\mathbf{P}(d(\xi_1,\ldots,\xi_k)\ge r)$ — вероятность того, что минимальное расстояние между парами различных точек ξ_i $(i=1,2,\ldots,n)$ не меньше r.

Если считать, что библиотеки V получаются случайным независимым бросанием k точек на отрезок $[0,1]$, где вероятность попадания в любую пару отрезков одинаковой длины одинакова, то $f(k,r)$ равна доле множества k-элементных библиотек, у которых минимальное расстояние между любыми двумя точками не меньше чем r, по отношению к множеству всех библиотек мощности k.

Справедлива следующая теорема.

Теорема 2.

$$f(k,r)=\begin{cases}1 & \text{если } r<0\\ (1-(k-1)\cdot r)^k & \text{если } 0\le r\le 1/(k-1)\\ 0 & \text{если } r>1/(k-1).\end{cases}$$

Доказательство:

Пусть l — вещественное число из отрезка $[0,1]$, r — вещественное число, k — натуральное число, большее 1, n — такое натуральное число, что $1\le n\le k$. Пусть $x_1,x_2,\ldots,x_k$ — независимые равномерно распределенные на отрезке $[0,1]$ случайные величины. Обозначим через $B(n,r,l)$ событие, что точки $x_1,x_2,\ldots,x_n$ попадают в отрезок $[0,l]$, и минимальное расстояние между парами различных точек x_i $(i=1,2,\ldots,n)$, не меньше r. Обозначим через $f(n,r,l)=\mathbf{P}(B(n,r,l))$. Понятно, что если $r<0$, то $f(n,r,l)=l^n$, а при $r>l/(n-1)$, $f(n,r,l)=0$. Поэтому мы будем рассматривать только случай, когда $0\le r\le l/(n-1)$.

Лемма 1. *Если* $0\le r\le l/(n-1)$, *то* $f(n,r,l)=(l-(n-1)\cdot r)^n$.

Доказательство будем вести индукцией по n.

Базис индукции. $n=2$.

Поскольку возможны два равновероятных события: случай когда $x_1 < x_2$, и когда $x_2 < x_1$, то достаточно рассмотреть первую ситуацию и удвоить полученный результат. Поскольку в этом случае x_1 может меняться от 0 до $l - r$, а x_2 — от $x_1 + r$ до l, то

$$f(2, r, l) = 2 \int_0^{l-r} dx_1 \int_{x_1+r}^{l} dx_2 = 2 \int_0^{l-r} (l - x_1 - r) dx_1 = (l - r)^2.$$

Индуктивный переход. Пусть утверждение леммы справедливо для любого натурального $q < n$ и любого вещественного $l \in [0, 1]$.

Через A_i обозначим событие, что случайная величина x_i, максимальна среди величин $x_1, \ldots, x_n$, здесь $i = 1, \ldots, n$. Понятно, что если $i, j \in \{1, \ldots, n\}$ и $i \neq j$, то $A_i \cap A_j = \emptyset$, кроме того $\mathbf{P}(A_i) = 1/n$, для любого $i \in \{1, \ldots, n\}$.

Легко видеть, что

$$\mathbf{P}(B(n, r, l)) = \sum_{i=0}^{n} \mathbf{P}(A_i \cap B(n, r, l)) = n \cdot \mathbf{P}(A_n \cap B(n, r, l)).$$

Поскольку в случае события $A_n \cap B(n, r, l)$ величина x_n может меняться от $(n - 1)r$ до l, а остальные $n - 1$ величины располагаются на отрезке $[0, x_n - r]$ и должны находиться на расстоянии не менее r, то согласно предположению индукции

$$\begin{aligned} f(n, r, l) &= \mathbf{P}(B(n, r, l)) = n \cdot \mathbf{P}(A_1 \cap B(n, r, l)) = \\ &= n \int_{(n-1)r}^{l} \mathbf{P}(B(n - 1, r, x_n - r)) dx_n = \\ &= n \int_{(n-1)r}^{l} (x_n - r - (n - 2)r)^{n-1} dx_n = \\ &= (l - (n - 1)r)^n. \end{aligned}$$

Тем самым лемма доказана.

Чтобы убедиться в справедливости утверждения теоремы 2 достаточно заметить, что $f(k, r) = f(k, r, 1)$.

Тем самым теорема 2 доказана.

Следующая теорема описывает асимптотическое поведение функции $f(k, r)$.

Теорема 3. *Пусть r_k — последовательность вещественных чисел, такая, что $0 \le r_k \le 1/(k-1)$. Тогда*

$$\lim_{k\to\infty} f(k, r_k) = \begin{cases} 0, & \text{если } 1/r_k = \bar{o}(k^2) \\ e^{-1/c}, & \text{если } 1/r_k \sim c \cdot k^2, \text{ где } c = const \\ 1, & \text{если } k^2 = \bar{o}(1/r_k). \end{cases}$$

Доказательство: Пусть $\alpha_k = \bar{o}(k)$ при $k \to \infty$. Воспользовавшись вторым замечательным пределом, легко получить

$$\begin{aligned} \lim_{k\to\infty} \left(1 - \frac{\alpha_k}{k}\right)^k &= \lim_{k\to\infty} \left(\frac{k}{k-\alpha_k}\right)^{-k} = \\ &= \lim_{k\to\infty} \left(\left(1 + \frac{\alpha_k}{k-\alpha_k}\right)^{\frac{k-\alpha_k}{\alpha_k}} \right)^{-\frac{\alpha_k k}{k-\alpha_k}} = \\ &= \lim_{k\to\infty} e^{-\frac{k\alpha_k}{k-\alpha_k}} = \lim_{k\to\infty} e^{-\alpha_k}. \end{aligned} \tag{5}$$

Рассмотрим случай, когда $1/r_k = \bar{o}(k^2)$ при $k \to \infty$.

Это означает, что для некоторой последовательности $\alpha_k \to \infty$ при $k \to \infty$ выполняется $r_k = \alpha_k/k^2$. Поскольку $r_k \le 1/(k-1)$, то достаточно рассмотреть два подслучая: $\alpha_k \sim c \cdot k$, где c — константа не превышающая 1, и $\alpha_k = \bar{o}(k)$. В первом подслучае

$$f(k, r_k) = \left(1 - \frac{(k-1)\alpha_k}{k^2}\right)^k \sim \left(1 - \frac{c(k-1)k}{k^2}\right)^k = \bar{o}(1).$$

Так как во втором подслучае $(k-1)\alpha_k/k = \bar{o}(k)$, то согласно (5)

$$\begin{aligned} \lim_{k\to\infty} f(k, r_k) &= \lim_{k\to\infty} \left(1 - \frac{(k-1)\alpha_k}{k^2}\right)^k = \\ &= \lim_{k\to\infty} e^{-\frac{(k-1)\alpha_k}{k}} = \lim_{k\to\infty} e^{-\alpha_k} = 0. \end{aligned}$$

Рассмотрим случай, когда $1/r_k \sim ck^2$ при $k \to \infty$, где $c = const$.

Поскольку $(k-1)/ck = \bar{o}(k)$, то согласно (5)

$$\lim_{k\to\infty} f(k, r_k) = \lim_{k\to\infty} \left(1 - \frac{k-1}{ck^2}\right)^k = \lim_{k\to\infty} e^{-\frac{k-1}{ck}} = e^{-1/c}.$$

И наконец, рассмотрим случай, когда $k^2 = \bar{o}(1/r_k)$ при $k \to \infty$.

Это означает, что для некоторой последовательности $\alpha_k \to 0$ при $k \to \infty$ выполняется $r_k = \alpha_k/k^2$. Поскольку $(k-1)\alpha_k/k = \bar{o}(k)$, то согласно (5)

$$\begin{aligned}\lim_{k\to\infty} f(k, r_k) &= \lim_{k\to\infty} \left(1 - \frac{(k-1)\alpha_k}{k^2}\right)^k = \\ &= \lim_{k\to\infty} e^{-\frac{(k-1)\alpha_k}{k}} = \lim_{k\to\infty} e^{-\alpha_k} = 1.\end{aligned}$$

Тем самым теорема 3 доказана.

Отсюда следует, что если в нашем распоряжении имеется объем памяти размера $c \cdot k^2$, то доля библиотек мощности k, для которых описанным алгоритмом мы можем находить ответ за 6 элементарных операций, равна $e^{-1/c}$. А если в нашем распоряжении имеется объем памяти больший по порядку, чем k^2, то для почти всех библиотек мы можем находить ответ за 6 элементарных операций. С другой стороны, если у нас имеется объем памяти, меньший по порядку, чем k^2, то почти всегда мы не сможем воспользоваться описанным выше алгоритмом поиска.

Обозначим через $\overline{d}(k)$ среднее значение описанной выше величины $d(\xi_1, \ldots, \xi_k)$, тогда справедливо следующее утверждение.

Теорема 4. $\overline{d}(k) = 1/(k^2 - 1)$.

Доказательство: Обозначим через $F(x)$ функцию распределения случайной величины $d(\xi_1, \ldots, \xi_k)$.

$$F(x) = \mathbf{P}(d(\xi_1, \ldots, \xi_k) < x) = 1 - \mathbf{P}(d(\xi_1, \ldots, \xi_k) \geq x) = 1 - f(k, x).$$

Тогда так как при $x \leq 0$ $F(x) = 0$, а при $x \geq 1/(k-1)$ $F(x) = 1$, то используя формулу интегрирования по частям, нетрудно получить

$$\begin{aligned}\overline{d}(k) &= \int_{-\infty}^{\infty} x dF(x) = \int_0^{\frac{1}{k-1}} x dF(x) = \\ &= xF(x)\Big|_0^{\frac{1}{k-1}} - \int_0^{\frac{1}{k-1}} F(x)dx = \\ &= \int_0^{\frac{1}{k-1}} f(k, x)dx = \int_0^{\frac{1}{k-1}} (1 - (k-1)x)^k dx = \frac{1}{k^2 - 1}.\end{aligned}$$

Тем самым теорема 4 доказана.

Отсюда следует, что в типичной ситуации достаточно иметь k^2 памяти, чтобы обеспечить время поиска в 6 элементарных операций.