Домашнее Задание по алгоритмам №5

Павливский Сергей Алексеевич , 873

25.02.2019

Задача 1.

Дано n слов длины k, состоящих из маленьких букв латинского алфавита. Предложите эффективный алгоритм их сортировки в лексикографическом (словарном) порядке.

Решение

Используем radix sort, сортировке по старшим разрядам (MSN). Такая сортировка будет устойчива, и будет работать за $O(n \times k)$

Док-во корректности: очевидно

Док-во эффективности: любая сортировка сравнением работает не менее, чем за $O(n \log n)$. В данном случае $k = const$, и можно считать сложностью $O(n \times m) = O(Cn) = O(n)$, т.е. такая сортировка будет эффективна в приведённом случае.

Задача 2.

Пусть числовой массив $a[1], \dots, a[n]$ строго унимодален на максимум. Это означает, что существует t, такое что

$$a[1] < a[2] < \cdots < a[t] > a[t+1] > \cdots > a[n-1] > a[n], \quad 1 \leq t \leq n.$$

1. Разрешается за один ход спросить значение одного элемента массива. Докажите, что можно найти значение максимального элемента $a[t]$ за не более $O(\log n)$ ходов.

Решение

На каждом ходе рассматриваемый отрезок делится попалам. Средний элемент ($\lceil \frac{n}{2} \rceil = m$) сравнивается с соседним: ($a[m] \vee a[m+1]$)

Если $a[m] < a[m+1]$, то max находится в массиве $a[m+1], ..., a[n]$, который станет следующим рассматриваемым отрезком.

Если $a[m] > a[m+1]$, то max находится в массиве $a[1], ..., a[m]$, который станет следующим рассматриваемым отрезком.

Таким образом, дойдём до момента, когда из элементов однозначно можно будет выделить максимум (когда из 2-х элементов одним сравнением можно будет выделить больший элемент — искомый максимум).

Таким образом, за $O(\log n)$ будет найден максимальный элемент.

Задача 3.

Имеется n монет, среди которых одна фальшивая, и чашечные весы. Настоящие монеты все имеют одинаковый вес, а фальшивая легче. На каждую чашку весов можно класть произвольное количество монет. Докажите, что фальшивую монету можно найти за $\log_3 n + c$ взвешиваний.

Решение

На каждом ходу рассматриваемое кол-во монет делится на 3 части, 2 из которых взвешиваются.

Пусть масса 1, 2 и 3 частей — a, b, c. a и b взвешиваются.

Если a = b, то фальшивая монета в c.

Если a < b, то фальшивая монета в a.

Если a > b, то фальшивая монета в b. Далее рассматриваем "кучу"с фальшивой монетой.

Таким образом доходим до этапа, когда количества может ≤ 3, когда можно будет однозначно сказать, которая монета — фальшивая.

Таким образом фальшивую монету можно найти за $\log_3 n + C$ взвешиваний.

Задача 4.

Докажите, что в условиях предыдущей задачи для нахождения фальшивой монеты необходимо $\log_3 n + c$ взвешиваний.

Решение

В любом случае на весы кладутся 2 кучки (случай взвешивания одной кучи, когда одна из частей пустая, в условиях данной задачи не даёт новой информации, поэтому не рассматривается). Взвешивать 2 неравные кучи смысла не имеет, так как, очевидно, куча с большим количеством монет будет весить больше (т.к. у фальшивая монета имеет положительную массу, меньшую массы настоящей).

При взвешивании можно определить, в какой из взвешиваемых куч находится фальшивая монета, или, что фальшивая монета находится ни в одной из этих куч. Поэтому в общем случае рассматриваемое количество монет уменьшается не более, чем в 3 раза (хотя бы в одной куче всегда есть $\frac{1}{3}$ от общего количества).

Таким образом, для нахождения фальшивой монеты необходимо $\log_3 n + c$ взвешиваний.

Задача 5.

Даны два отсортированных массива длины n. Предложите как можно более эффективный алгоритм поиска медианы в массиве, состоящем из всех данных $2n$ элементов. Можно считать, что все элементы различные. Докажите корректность алгоритма и оцените его сложность (количество сравнений). В этой задаче обращения к элементам массива выполняются за O(1), читать оба массива целиком не требуется, считайте, что они уже лежат в памяти.

Решение

Пусть изначальные массивы длины n — A и B, а массив, состоящий из всех данных $2n$ элементов — S.

Т.к. A и B отсортированы, то медиану в них можно найти за $O(1)$. Пусть медиана A равна m_1, медиана B равна m_2, медиана S — m.

По условию задачи можем считать, что все элементы различны, поэтому $m_1 \neq m_2$.

Если $m_1 > m_2$, то $m_2 \leq m \leq m_1$ (все элементы массива A "правее" m_1, по значению больше его, и больше, чем все предыдущие до m_1, а, значит, не могут быть медианой S, т.к. находятся во второй половине отсортированного S).

Если $m_2 > m_1$, то $m_1 \leq m \leq m_2$ (аналогично все элементы массива A "правее" m_1, по значению больше его, и больше, чем все предыдущие до m_1, а, значит, не могут быть медианой S, т.к. находятся во второй половине отсортированного S).

Таким образом мы установили, что медиана массива S находится между медианами массивов A и B. Далее, A_1 становится массив из элементов A, больших m_1, а B_1 — из элементов массива B, больших m_2, и операция повторяется для A_1 и B_1. Таким образом, процесс повторяется до той поры, пока однозначно нельзя назвать медиану полученных массивов (пока количество элементов не станет меньше или равно 3).

<u>Доказательство корректности</u> : На каждом шаге рассматриваемое множество содержит медиану массива, поэтому финаль-

ный шаг будет также содержать медиану.

Мощность множества на каждом шаге уменьшается вдвое, а значит количество рассматриваемых аргументов будет уменьшаться. Алгоритм закончит работу, когда однозначно можно будет назвать медиану полученных массивов (пока количество элементов не станет равным 2 или 3).

Пусть оставшиеся 3 элемента — a, b и c (или a и b). В силу того, каким образом мы задали алгоритм, a $<$ b $<$ c (или a $<$ b). Тогда медиана будет очевидно равна b (в случае с двумя элементами, считаем медиану массива из n чисел значением $\lceil \frac{n}{2} \rceil$ элемента.

Оценка сложности : На каждом шаге количество рассматриваемых элементов уменьшается вдвое, поэтому всего таких шагов будет сделано $\log n + c$, т.е. сложность алгоритма можно оценить как $O(\frac{n}{2})$

Задача 6.

Определите, что число является значением данного многочлена с натуральными коэффициентами в натуральной точке. На вход задачи подаются натуральные числа $n, a_0, \ldots a_n$ и y. Необходимо определить, существует ли натуральное число x, такое что

$$y = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0.$$

Решение

Подставим в уравнение $x_1 = 0$ и $x_2 = \frac{y - a_0}{a_n + a_{n-1} + \ldots + a_1}$. Если разности $(y(x_1) - y)$ и $(y(x_2) - y)$ разного знака , то корень есть , иначе его нет . В случае если корень есть , то каждом шаге вычисляем $x_3 = x_2 - x_1$. Если $y(x_3) > y$ и $y(x_1) > y$, то $x_1 = x_3$. Иначе $x_2 = x_3$. Так до тех пор пока $(y(x_3) <> y$, или $(y(x_3) - y) > e$, где e стремится к 0 . Если достигается первое , то мы нашли натуральный корень , иначе корень не натуральный , и натурального корня не существует .

Задача 7.

Есть n монет разного веса. За одно взвешивание можно сравнить по весу любые две монеты.

1. Найдите самую тяжёлую и самую лёгкую за $\frac{3}{2}n + O(1)$ взвешиваний.

Решение

Разобьём изначальную "кучу"размера n по парам. В каждой паре проведём сравнение между её элементами. Элементы, "победившие"в сравнении, вес которых оказался больше, выделим в одну кучу (назовём её кучей 1), "проигравшие— в другую (назовём её кучей 2). Таким образом получим 2 кучи размерами $\frac{n}{2}$.

Монета с наибольшим весом не должна проиграть ни в одном сравнении, поэтому достоверно можно сказать, что она находится в 1 куче. Монета же с наименьшим весом, напротив, должна проиграть в каждом сравнении, поэтому находится во 2 куче.

Сравнений на данном этапе проведено — $\frac{n}{2}$.

На следующем шаге разобьём каждую из куч по парам таким же образом. Аналогично проведём сравнение между элементами пар. Элементы из 1 кучи, "победившие"в сравнении, выделим в кучу 1.1, проигравшие — в кучу 1.2. Аналогично выделим элементы со 2 кучи: "победившие"в 2.1, "проигравшие— в 2.2.

По тем же рассуждениям монета с наибольшим весом находится в куче 1.1, с наименьшим — в 2.2.

Дальнейшее рассмотрение куч 1.2 и 2.1 не имеет смысла для решения задачи, т.к. ни самая тяжёлая, ни самая лёгкая монеты не могут находится в этих кучах.

Сравнений на данном этапе проведено — $\frac{n}{2} + 2 \cdot \frac{n}{4} = n$.

Далее таким же образом производим разбиение куч: 1.1 и 2.2, 1.1.1 и 2.2.2 и так далее.

Число сравнений будет составлять сумму $\frac{n}{2}$ и сумму геометрической прогрессии с шагом $\frac{1}{2}$ и начальным элементом $\frac{n}{2}$:

$\frac{n}{2} + \frac{n}{2} + ... = \frac{3}{2}n$.

Тогда самую тяжёлую и самую лёгкую монету можно будет найти за $\frac{3}{2}n + O(1)$.