Алгоритмы. ДЗ на неделю 5.

Прохоров Юрий, 771

Задача 6 (с семинара)

Такого алгоритма не существует. Допустим, он есть и решил требуемую задачу, узнав при этом значения $n-1$ бита. Рассмотрим 3 случая.

1. Алгоритм не спросил про крайний бит слева. Тогда на входах

$$[0,1,1,\ldots,1] \qquad \text{и} \qquad [1,1,1,\ldots,1],$$

вследствие своей детерминированности, алгоритм вернем одно и то же значение, хотя ответ разный.
2. Алгоритм не спросил бит в середине строки. На входах

$$[1,\ldots,1,0,1,\ldots,1] \qquad \text{и} \qquad [1,\ldots,1,1,1,\ldots,1]$$

будут одинаковые ответы, что неверно.
3. Алгоритм не спросил про крайний бит справа. Аналогично первому случаю, достаточно рассмотреть входы

$$[0,\ldots,0,0,1] \qquad \text{и} \qquad [0,\ldots,0,0,0].$$

Доказательство, что получить ответ невозможно меньше, чем за $n-1$ вопросов, абсолютно аналогичное.

Задача 7(б) (с семинара)

Решение для пункта (a) аналогично, только немного упрощено.

Взвешиваем $[1,2,3,4] \times [5,6,7,8]$. Рассмотрим три случая, два из которых аналогичны.
1. Равенство. Тогда монеты $1,2,3,4,5,6,7,8$ — нормальные. Взвесим $[1,2,3] \times [9,10,11]$.
  1.1. Равенство. Тогда ответ — 12.
  1.2. Меньше. Тогда фальшивая монета тяжелее остальных. Взвесим $[9] \times [10]$.
    1.2.1. Равенство. Ответ — 11.
    1.2.2. Меньше. Ответ — 10.
    1.2.3. Больше. Ответ — 9.
  1.3. Больше. Аналогично пункту 1.2.
2. Меньше. Тогда монеты $9,10,11$ — нормальные. Взвесим $[1,2,5] \times [3,4,9]$.
  2.1. Равенство. Тогда фальшивая монета среди монет $6,7,8$. Монеты $1,2,3,4$ — нормальные, значит, фальшивая монета тяжелее. Далее аналогично пункту 1.2.
  2.2. Меньше. Пусть 5 — фальшивая монета, следовательно, она легче остальных. Тогда самое первое взвешивание показало бы больше. Тогда одна из монет $1,2,3,4$ — фальшивая. Из самого первого взвешивания следует, что она легче. Тогда одна из монет $1,2$ — фальшивая. Взвесим $[1] \times [2]$.
    2.2.1. Меньше. Ответ — 1.
    2.2.2. Больше. Ответ — 2.
  2.3. Больше. Пусть фальшивая монета среди монет $1,2$, тогда она тяжелее остальных, и первое взвешивание дало бы больше. Тогда либо 5 — тяжелая фальшивая монета, либо одна из $3,4$ — легкая фальшивая монета. Взвесим $[3] \times [4]$.
    2.3.1. Равенство. Ответ — 5.
    2.3.2. Меньше. Ответ — 3.
    2.3.3. Больше. Ответ — 4.
3. Больше. Аналогично пункту 2.

Задача 1 (ДЗ)

Заменим каждый символ в строках на их номер в латинском алфавите мощности $N = 23$. Тогда получим n чисел длины k в системе счисления с основанием N. Максимальное значение каждого знака ограничено маленьким числом (по сравнению с n и k), поэтому алгоритм сортировки $RadixSort$ отсортирует данный массив за время $\Theta(nk)$, то есть за линейное относительно длины входа число операций.

Задача 2(1) (ДЗ)

Узнаем значение медианы $a[m]$ массива, элемента слева и справа от нее. Рассмотрим 3 случая:

1. $a[m-1] < a[m] < a[m+1]$. Тогда ищем максимум в правой половине массива.
2. $a[m-1] < a[m] > a[m+1]$. Максимум найден.
3. $a[m-1] > a[m] > a[m+1]$. Ищем максимум в левой половине массива.

В худшем случае число ходов:

$$F(n) = F\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 3 \qquad \Longrightarrow \qquad F(n) = O(\log n)$$

Задача 3 (ДЗ)

Пусть число монет $n = 3^k$. Иначе, добавим нормальных монет до степени тройки, при этом число операций уменьшиться не может.

Сначала взвесим монеты с номерами $\left[1, \ldots, \frac{n}{3} - 1\right]$ и $\left[\frac{n}{3}, \ldots, \frac{2n}{3} - 1\right]$. Если весы показали равенство, то фальшивая монета в оставшейся куче, если меньше — то в первой куче, если больше — то во второй. Когда останется 3 монеты, то на определение фальшивой требуется одно взвешивание. Тогда число взвешиваний:

$$F(n) = F\left(\frac{n}{3}\right) + 1 = F\left(\frac{3^k}{3^2}\right) + 2 = F\left(\frac{3^k}{3^{k-1}}\right) + k - 1 = k = \log_3 n.$$

Задача 4 (ДЗ)

Заметим, что если мы кладем на чаши весов разное число монет, то от результата взвешивания мы не получаем никакой информации, поэтому такие взвешивания мы рассматривать не будем.

Пусть на первом шаге мы взвесили две группы монет по k штук:

$$\underbrace{\bigcirc\ \bigcirc \ldots \bigcirc}_{k \text{ монет}} \qquad \underbrace{\bigcirc\ \bigcirc \ldots \bigcirc}_{k \text{ монет}} \qquad \underbrace{\overline{\bigcirc}\ \overline{\bigcirc} \ldots \overline{\bigcirc}}_{n-2k \text{ монет}}$$

После взвешивания в худшем случае мы узнаем, что монета находится в группе из $\max(k, k, n-2k) \geq \frac{n}{3}$ монет. Случай равенства был рассмотрен в задаче 3, поэтому за $\log_3 n$ найти фальшивую монету можно. Если знак в неравенстве строгий, то число монет в задаче уменьшается медленнее, поэтому меньше ходов сделать невозможно.

Задача 5 (ДЗ)

Медианой будем считать порядковую статистику под номером $\left\lfloor \frac{n+1}{2} \right\rfloor$. Все элементы в массивах различны.

Алгоритм

Построим рекурсивный алгоритм нахождения медианы.

В исходных массивах A и B выберем медианы $m_1 = A[p]$ и $m_2 = B[q]$. Пусть для определенности $m_1 < m_2$. Отбросим все элементы первого массива, которые строго меньше m_1, и все элементы второго массива, которые строго больше m_2. Первым массивом (массивом A) называется массив с меньшей медианой, а вторым (B) — с большей медианой. Далее запустим этот же алгоритм на входах $A[p, \dots, n]$ и $B[1, \dots, q]$.

Когда в каждом из массивов останется по одному элементу, просто выберем из них наименьший.

Function `Median`$(A[1, \dots, n], B[1, \dots, n])$ **:**
  if $A.Length == B.Length == 1$ **then**
    Output: $\min(A[1], B[1])$
  $p = \left\lfloor \frac{A.Length+1}{2} \right\rfloor$;
  $q = \left\lfloor \frac{B.Length+1}{2} \right\rfloor$;
  $m_1 = A\left[p\right]$;
  $m_2 = B\left[q\right]$;
  if $m_1 < m_2$ **then**
    Output: `Median`$(A[p, \dots, n], B[1, \dots, q])$
  else
    Output: `Median`$(B[q, \dots, n], A[1, \dots, p])$
  end
end

Корректность

Достаточно показать, что после каждого рекурсивного вызова медиана исходного массива остается в одном из массивов.

На первом шаге (пусть $m_1 < m_2$) отбрасывается $\left\lfloor \frac{n-1}{2} \right\rfloor$ элементов A. Рассмотрим произвольный из них — элемент X. Элемент X меньше любого элемента правее медианы A и самой медианы — в сумме $\left\lceil \frac{n+1}{2} \right\rceil$ элементов. Из неравенства $m_1 < m_2$ следует, что X также меньше правой половины массива B — еще $\left\lceil \frac{n+1}{2} \right\rceil$ элементов. Таким образом, отброшенный элемент X меньше, чем как минимум $2\left\lceil \frac{n+1}{2} \right\rceil > n$ элементов общего (соединенного) массива, поэтому X не является его медианой. Аналогично для отброшенных элементов массива B.

Покажем, что на следующем шаге медиана опять сохранится в массивах. Найдем в обрезанных массивах $\widetilde{A}$ и $\widetilde{B}$ медианы $\widetilde{m_1}$ и $\widetilde{m_2}$. Пусть вновь $\widetilde{m_1} < \widetilde{m_2}$ для определенности. Будем опускать округления, так как они в данном случае аналогичны. Отбрасываем от массива $\widetilde{A}$ еще $\frac{n}{4}$ элементов, меньших медианы $\widetilde{m_1} = kStat\left(\frac{3n}{4}, A\right)$ — порядковой статистики с номером $\frac{3n}{4}$ исходном массиве A. Тогда новые отброшенные элементы меньше $\frac{n}{4}$ элементов из массива A и также меньше $\frac{3n}{4}$ элементов из массива B, так как они меньше $\widetilde{m_2} = kStat\left(\frac{n}{4}, B\right)$. В сумме они меньше n элементов, поэтому медианой не являются. Аналогично с другой частью отброшенных элементов.

Продолжая аналогичные рассуждения, убедимся, что медиану сцепленных исходных массивов мы не теряем ни на одном шаге. А на последнем шаге она определяется, как наименьший из двух оставшихся, что удовлетворяет введенному определению медианы.

Оценка по времени

Время работы алгоритма при достаточно n:

$$T(2n) = T\left(2\left\lceil \frac{n+1}{2} \right\rceil\right) + c \le T\left(n+2\right) + c,$$

$$T(2n) \le T(2n+4) \le T\left(n+2\right) + c \qquad \Longrightarrow \qquad T(2n) = O(\log(n+2)) = O(\log n).$$

Задача 7(1) (ДЗ)

Устроим между монетами турнир со стандартной сеткой на выбывание. Таким образом, найдем самую тяжелую монету. Получим дерево с $n-1$ ребрами, отвечающими за взвешивания. Между теми монетами, которые проиграли в первом раунде (их ровно $\lceil \frac{n}{2} \rceil$) устроим еще один турнир, но теперь будем искать самую легкую. Это дерево будет состоять из $\lceil \frac{n}{2} \rceil - 1$ ребра.

Таким образом, на решение задачи потребуется $\frac{3n}{2} + c$ взвешиваний.