

Домашнее задание 5

Христолюбов Максим, 771

Задание 6 с семинара

Нет, не возможно. Допустим алгоритм проверил все биты кроме хотябы одного. В худшем случае или слева от не проверенного бита будет стоять 0, либо справа 1. Тогда для случая, когда непроверенный бит - 1 и для случая с 0 алгоритм получил одинаковый набор данных, а поскольку алгоритм детерминирован, то он всегда будет давать один ответ, т.е. ошибаться в одном из случаев.

Задание 7 с семинара

а) Пронумеруем монеты от 1 до 10. Сравним группы из монет 1, 2, 3 и 4, 5, 6.

Если весы покажут равенство, тогда фальшивая монета в остальных четырех. Сравним 7, 8, 9 и заведомо настоящие 1, 2, 3.

Если весы покажут равенство, то фальшивая монета - последняя.

Если нет весы покажут, что 7, 8, 9 тяжелее значит, фальшивая монета тяжелее остальных. Сравним 7 и 8, в случае равенства 9 - фальшивая, иначе фальшивая более тяжелая из 7 и 8.

Если весы покажут, что 1, 2, 3 тяжелее чем 4, 5, 6, то будем делать следующие шаги. В случае когда они будут легче из симметричности потребуется произвести те же действия, поменяв кучки местами.

Сравним 1, 4 и 2, 3.

Если весы показали равенство, то фальшивая монета среди 5, 6, причем она легче остальных. Сделаем последнее сравнение и найдем фальшивку.

Если весы показали, что 1, 4 тяжелее, то сравним 1 и заведомо настоящей 7. Если они равны, то 4 - фальшивая, иначе 1 - фальшивка.

Если показали что 2, 3 тяжелее, то фальшивая - самая тяжелая из них.

Задание 1

Можно отсортировать слова сначала по последнему символу устойчивой сортировкой. Это можно сделать за $\Theta(n)$, так как кол-во латинских символов константа. Занумеруем буквы, запишем в элементы массива C сумму количеств каждого из символов, номер которого меньше чем номер элемента в массиве C , а потом пройдемся по словам и поставим на C_p место p -ый символ, $C_p = C_p + 1$ за $O(n)$. То есть воспользуемся RadixSort, потребуется k сортировок по $\Theta(n)$ операций - $\Theta(nk)$.

Задание 2

1) Спросим сначала чему равен $\frac{n}{3}$ и $\frac{2n}{3}$ элементы. Если $\frac{2n}{3}$ больше, то это значит что t -ый элемент не может лежать в первой трети массива - левее $\frac{n}{3}$ -ого элемента, если $\frac{n}{3}$ больше, то искомый элемент не лежит в последней трети массива. Повторим операцию с частью массива длины $\frac{2n}{3}$, где может находится искомый элемент. На каждом шаге массив уменьшается в $\frac{2}{3}$ раз, значит за $\log_{\frac{2}{3}} n$ шагов найдется искомый элемент, сложность - $\Theta(\log n)$.

Задание 3

Разделим монеты на 3 части, по крайней мере 2 из которых содержат $\lfloor \frac{n}{3} \rfloor$ кол-во монет. Сравним на весах эти 2 равные по кол-ву монет части. Если они равны, то фальшивая монета находится в остальных монетах, если одна куча перевесила, то фальшивая монета находится в более легкой куче. Повторим операцию для той кучи, где находится фальшивая монета. На каждом кол-во монет уменьшается в $\frac{2}{3}$ раз, следовательно за $\log_3 n + 1$ взвешиваний мы дойдем до кучи из одной или 2 монет, в которой можно найти фальшивую за 0 или 1 взвешивание. Значит, можно сделать $\log_3 n + 1$ операций и найти фальшивую монету.

Задание 4

Одно взвешивание дает информацию о принадлежности фальшивой монеты одному из 3 множеств (множества зависят от того какие имеено кучи монет взвешивать). Для того чтобы алгоритм мог определить фальшивую монету, ни для каких 2 монет результаты взвешиваний не должны быть одинаковыми, иначе в силу детерминированности алгоритма он

будет ошибаться в одном из случаев. Запишем результаты взвешиваний как троичные числа, в которых k -ый разряд будет говорить в какой из 3 частей лежит монета. Тогда для каждых 2 различных монет им соответствующие троичные числа должны быть разными. Троичных чисел длины k - $3^k \geq n$, значит $k \geq \log_3 n$, следовательно требуется произвести не менее $\log_3 n$ взвешиваний.

Задание 5

Сравним элементы, лежащие на $\frac{n}{2}$ местах в массивах - это медианы, поскольку массивы упорядочены. Пусть для определенности $A[\frac{n}{2}] > B[\frac{n}{2}]$ (если $A[\frac{n}{2}] = B[\frac{n}{2}]$, то это и есть медиана, она найдена). Тогда общая медиана лежит в первой половине массива A или во второй половине массива B , так как значение общей медианы лежит между значениями медиан массивов. Возьмем за исходные массивы соответствующие половины и повторим операцию. На каждом шаге область поиска медианы уменьшается в 2 раза, значит через $\log_2 n$ шагов останется один элемент - общая медиана. На каждом шаге делается фиксированное кол-во шагов, сложность $\Theta(\log n)$.

Задание 6

Будем перебирать числа, начиная с 1 и считать значение, пока значение многочлена не будет больше или равно y . В первом случае такого числа x не существует, во втором - число x , на котором остановился алгоритм - искомое. Сложение n чисел займет $\Theta(n)$ действий, многочлен станет больше или равен y через $\Theta(\sqrt[n]{y})$ шагов, сложность алгоритма - $\Theta(n \sqrt[n]{y})$.

Задание 7

1) Разобьем монеты на пары и сравним их попарно - $\frac{n}{2}$ взвешиваний, те что окажутся тяжелее отложим в первую кучу, те что легче - во вторую. Очевидно самая тяжелая находится в первой, а самая легкая - во второй. Теперь взвесим попарно монеты из первой кучи ($\frac{n}{4}$ взвешиваний) и отбросим легкие, а с тяжелыми повторим то же самое - будем попарно сравнивать и отбрасывать легкие пока не останется единственная монета - самая тяжелая. Кол-во действий будет равно $\frac{n}{4} + \frac{n}{8} + \dots + \frac{n}{p} \leq \frac{n}{2}$. Со второй кучей будем делать то же самое, только отбрасывать тяжелые и в

итоге получим самую легкую, аналогично сделав не более $\frac{n}{2}$ сравнений. Всего $\frac{3n}{2} + c$ сравнений, константа c возникает из-за того что не всегда монеты будут делиться ровно пополам.