

1.

Обозначим массив слов $a[n][k]$ – n слов длины k .

Разобьем слова на "разряды" (буквы) и будем сортировать эти "корзинки букв" по очереди начиная с первого разряда (буквы). Сортировка по каждой букве должна быть устойчивой: будем использовать сортировку подсчетом, так как число возможных значений ограничено маленьким числом (26 букв латинского алфавита), которая работает за линейное время. Таким образом, пройдя по всем разрядам (буквам), получим отсортированный массив слов. Сортировка букв проходит аналогично сортировке чисел с учетом порядка их следования в алфавите.

Корректность:

Производится устойчивая сортировка по каждой букве слов, начиная с первой, так что на выходе получается полностью отсортированный массив слов.

Сложность:

Всего выполняется k шагов (k раз сортировка подсчетом по каждой из k букв слова). Каждая такая сортировка подсчетом занимает $O(n)$ (что было показано на лекции). Таким образом, итоговая сложность алгоритма $O(n \cdot k)$.

2.

1. На первом шаге спросим значение среднего элемента $a[\lceil \frac{n}{2} \rceil]$. На втором значение следующего элемента $a[\lceil \frac{n}{2} \rceil + 1]$. Затем в зависимости от знака разности этих двух элементов перейдем либо в левую, либо в правую часть массива: если $a[\lceil \frac{n}{2} \rceil] - a[\lceil \frac{n}{2} \rceil + 1] > 0$, то следующий элемент необходимо брать средний из левой части, то есть, $a[\lceil \frac{n}{4} \rceil]$, иначе из правой части $a[\lceil \frac{n}{2} + \frac{n}{4} \rceil]$. Затем снова необходимо сравнить спрошенный элемент с $a[\lceil \frac{n}{2} \rceil]$. И снова повторить этот процесс. Таким образом, каждый раз уменьшая отрезок поиска в два раза, алгоритм спустится к одному элементу, который как раз и будет максимумом.

Корректность:

На каждом мы сравниваем два элемента из отрезка, чтобы определить характер монотонности на этом отрезке. Если элементы возрастают, следует смотреть в правой части ведь именно там должна пройти смена монотонности через максимальный искомый элемент. И, наоборот, если убывают из той же логики следует смотреть в левой части. Кроме первого раза, когда мы сравниваем соседние элементы, затем мы каждый раз искомый отрезок делим пополам, таким образом, спускась к одному элементу – максимуму массива, который расположен между двумя монотонными частями.

Сложность:

На каждом шагу, кроме первого, происходит деление подмассива пополам, так что высота дерева для этого алгоритма и соответственно сложность будет $O(\log n)$

2. Деление подмассива можно производить не пополам, а в золотом отношении. То есть, выбирать последовательно значения массива, с элементами, соответствующими делению подмассива в отношении $a = \frac{1+\sqrt{5}}{2}$. Однако на последнем шаге останется провести константное число операций, так как n не является степенью a . Так что итоговое число ходов будет $\log_a n + c$.

3.

Для начала заметим, что если у нас есть кучка из трех монет, то за одно взвешивание можно определить фальшивую монету: положим на весы две любые монеты, если они уравнились друг друга, то, значит, фальшивая – оставшаяся, иначе фальшивая – та, которая легче.

Тогда будем на каждом шаге делить кучку монет на три и сравнивать две из них. Согласно сказанному выше, за одно взвешивание можно определить кучку с фальшивой монетой и продолжить с ней те же операции деления на три и взвешивания.

Если n – степень 3, то на последнем шаге останется 3 монеты, иначе меньше 3 и это будет константное число операций. Таким образом, итоговое число операций будет $\log_3 n + c$.

4.

Покажем, что невозможно в кучке из 4 монет определить фальшивую за одно взвешивание гарантированно. Если мы разделим кучку на две части по две монеты, то точно в одной из них будет фальшивая, и эта кучка будет легче, но определить какая из двух без повторного взвешивания невозможно. И если невозможно определить фальшивую в кучке из 4 монет, то из большей уж точно. Таким образом, необходимо, чтобы деление было на 3. Таким образом, необходимо $\log_3 n + c$ операций.

5.

Обозначим массивы a и b . Нумеровать будем с 1. Медианой, согласно лекциям будем считать $\lceil \frac{n}{2} \rceil$ -статистику.

Пусть $m_1 = a[\lceil \frac{n}{2} \rceil]$, $m_2 = b[\lceil \frac{n}{2} \rceil]$ – медианы в двух массивах соответственно. Если $m_1 = m_2$, то возвращаем это значение $m_1 = m_2$. Если $m_1 > m_2$, то удалим в массиве a все элементы после $a[\lceil \frac{n}{2} \rceil]$, а в массиве b – до $\lceil \frac{n}{2} \rceil$. Иначе удаляем в массиве a все элементы до $a[\lceil \frac{n}{2} \rceil]$, а в массиве b – после $\lceil \frac{n}{2} \rceil$. Прodelываем этот процесс, пока размер подмассивов не станет равным двум, тогда медиану можно найти следующим образом: необходимо произвести слияние двух подмассивов из двух элементов. Пусть c – слитый массив, тогда $c[3]$ – искомая медиана. Так как элементов всего 4, то это требует константную сложность. Слияние производится как в быстрой сортировке, как это было показано на лекции.

Корректность:

Так как массивы отсортированные, то сравнив медианы двух массивов можно определить в каком подмассиве (правом или левом) находится их общая медиана. Ведь, если медиана первого массива больше, то уж точно их общая медиана не может лежать в части первого массива, где элементы больше этого числа или в части второго массива, где элементы лежат раньше его медианы, которая в свою очередь меньше медианы первого массива. Таким образом, каждый раз уменьшая диапазон поиска в два раза, мы дойдем до двух массивов из двух элементов, которые можно легко слить в единый массив и взять средний элемент.

Сложность:

На каждом шаге длина массива уменьшается в два раза. На последнем шаге производится слияние двух массивов из 4 элементов, что требует константную сложность. Таким образом, общая сложность: $O(\log n)$.

6.

Перенесем a_0 в другую сторону и посмотрим в каком диапазоне чисел следует искать значение аргумента.

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

Так как все числа натуральные, то можно записать следующие оценки:

$$y - a_0 = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x \leq (a_n + a_{n-1} + \dots + a_1) x^n,$$

$$y - a_0 = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x \geq (a_n + a_{n-1} + \dots + a_1) x$$

$$\text{Таким образом, } x \geq \sqrt[n]{\frac{y-a_0}{a_n+a_{n-1}+\dots+a_1}}, \quad x \leq \frac{y-a_0}{a_n+a_{n-1}+\dots+a_1}$$

Затем проходя по данному интервалу, только по целым точкам, подставляются значения аргумента и проверяется равенство. В худшем случае проверятся все значения из диапазона и выведется сообщение об отсутствии подходящего значения аргумента.

Корректность:

Получена теоретически оценка на возможный диапазон принимаемых значений аргументом и для каждого из них (в худшем случае) будет проверено равенство. Таким образом, пропусков в значениях аргумента нет, все значения будут просмотрены.

Сложность:

Границы диапазона вычисляются за константную сложность. В худшем случае будут пройдены все значения из диапазона и подставлены в равенство.

При каждой проверке равенства проводится $n+1$ умножение чисел и n сложений. Количество проходов равно количеству m целых точек в диапазоне $\left[\sqrt[n]{\frac{y-a_0}{a_n+a_{n-1}+\dots+a_1}}; \frac{y-a_0}{a_n+a_{n-1}+\dots+a_1} \right]$.

Таким образом, итоговая сложность: $O(n \cdot m)$

7.

Проведем следующие рассуждения:

Все этажи разделим на k групп. Первая, самая нижняя группа содержит h этажей. Бросаем шарик с верхнего этажа этой группы. Если он разбился – бросаем второй шарик с нижнего этажа этой группы, и последовательно повторяем броски, поднимаясь на один этаж. Если шарик с верхнего этажа не разбился – передвигаемся на верхний этаж второй группы. Одна попытка уже использована, поэтому во второй группе должно быть на один этаж меньше, т.е. $h-1$. Рассуждая аналогично, в третью группу включим $h-2$ этажа, в четвертую $h-3$ и т.д.

Таким образом, получим:

$$h + (h-1) + (h-2) + \dots + (h-k+1) = \frac{(2h-k+1)k}{2} = 100 \quad (\text{по формуле суммы арифметической прогрессии})$$

$$\text{Следовательно, } h = \frac{100}{k} + \frac{k}{2} - \frac{1}{2}$$

Найдем минимум этой функции: $\frac{-100}{k^2} + 1/2 = 0$ (приравняли производную к нулю)

Откуда получим: $k = \sqrt{200}$. Так как k – целое: $k = 14$, $h = \frac{100}{14} + \frac{14}{2} - \frac{1}{2}$. Таким образом, **$h = 14$** . Максимальное возможное количество бросаний равно, соответственно, тоже 14 (для любого этажа любой группы, вплоть до сотого).