Домашнее задание №3

Томинин Ярослав, 778 группа

4 марта 2018 г.

1.

Проделаем наш алгоритм 5=101=XXSX. Тогда y=x, потом $y = x^2$, потом $y = x^4$, потом $y = x^5$. Следовательно алгоритм выдаст 3^5

2.

Этот алгоритм реализует функцию возведения в степень. Здесь мы видим, что с помощью X мы делаем умножение степени на 2, а с помощью X добавляем к степени 1. Поэтому мы ноль заменяем только на X, а 1 меняем на SX. В начале мы убираем X, потому что нам не нужно увеличивать степень на 2 в первый раз.

3.

Допустим обратное: алгоритм некорректно возводит число в степень, это означает что в какой-то момент сложение степени сработало не правильно. Поймем, что если двоичная запись числа имеет k порядок, то наше число можно получить следующим образом: умножить 1 на 2^i, где i - количество X справа от S. А именно это и реализует наш алгоритм(грубо говоря мы раскладываем наше число по степеням двойки).

Сложность алгоритма: каждый раз мы возводим число y в квадрат (на некоторых шагах еще и умножение y на x). Эти действия мы делаем n раз. Таким образом, учитывая то, что арифметические операции стоят $O(1)$, получим сложность $O(n)$.

2.

Создадим новый массив, в котором на i месте будет записана разность $r_i - l_i$. Тогда после этого найдем $\frac{n}{3}$ и $\frac{n}{3} - 1$статистики для нашего массива. Понятно что это будет разность координат отрезков, которые являются $\frac{2n}{3}$ и $\frac{2n}{3} + 1$, если нумеровать отрезки по убыванию. Следовательно этот отрезок, являющийся разностью тих двух отрезков, и содержит все точки прямой, которые покрыты ровно $\frac{2n}{3}$. Сложность алгоритма O(n)

3.

Применим ту же логику. Выделим массивы по 7 элементов, выберем в каждом из них медиану, поом составим массив медиан и найдем из них медиану, раположим элементы в массив следующим образом: справа будут элементы, большие нашей медианы медиан, а слева меньшие. Тогда сделаем оценку, за i обозначим место нашей медианы медиан. Тогда $\frac{3n}{14} \leq p \leq \frac{11n}{14}$ и $\frac{3n}{14} \leq n - p \leq \frac{11n}{14}$

Сложность этого алгоритма $T(n) = T(\frac{11n}{14}) + T(\frac{n}{7} + cn)$

Сделаем индукционное предположение, что $T(n) = \Theta(n)$, тогда нам нужно доказать, что $\exists c_1 : c_1 n = \frac{13n}{14} + c$.

$$c_1 = \frac{14c}{n}$$

В силу того, что n может быть сколь угодно большим, делаем вывод, что

$T(n) = \Theta(n)$
4.
Проссумируем все элементы массива. Мы получим количество едениц, тогда запишем k едениц в конец массива, а оставшиеся места оставим в конце.
5.
Перенесем b в правую часть и найдем остаток при делении на M числа -b(Обозначим его за k). После этого найдем НОД(a,M). Это займет у нас $O(n^3)$, далее мы поймем делиться ли полученное число на k, проведем деление с остатком, это займет $O(n^2)$. После этого поймем, что если остаток равен 0, то у нас есть решение. Так как наша группа циклическая(элемент b принадлежит нашей циклической группе) и элемент a является образующим, его порядок равен M/НОД(a,M),это означает что нам достаточно O(n) операций, чтобы найти степень α, в которую надо возвести наш элемент a и получить k по mod M. Тогда результатом нашего алгоритма будет число x равное $[a](\alpha - 1)$.
6.
Поймем, чо если изначально наш массив отсортирован, то глубина стека будет равняться n, так как каждый раз наша функция будет вызывать массив, уменьшая количество элементов на 1. Следовательно сложность (n^2)
На каждом уровне рекурсии будем находить сначала медиану и брать ее за основной элемент, тогда количество элементов массива каждый раз будет уменьшаться в двое. Следовательно при любом раскладе глубина стека будет равна $\Theta(\log n)$
7.
Дли начала поймем, что в нашем алгоритме дваэлемента массива будут сравниваться только один раз, потому что опорный элемент может быть вызван только одной функцией сортировки. Определим велечину X_{ij}, которая будет говорить о том, сравнивались ли i и j элементы нашего массива. За X обозначим общее количество сравнений, тогда в силу того, что сравнение между элементами производится только один раз

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

Применим операцию мат. ожидания к обоим частям

$$E[X] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E\left[X_{ij}\right]$$

В общем случае ситуация такая. Поскольку предполагается, что значения всех элементов различаются, при выборе в качестве опорного элемента такого x, что $z_i < x < z_j$, элементы z_i иz_j впоследствии сравниваться не будут. С другой стороны, если в качестве опорного элемент z_i выбран до любого другого элемента Z_{ij}, то он будет сравниваться с каждым элементом множества Z_{ij}, кроме себя самого. Тогда вяроятность того что мы будем сравнивать i и j элементы нашего массива будет равна удвоенной вероятно

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E\left[\frac{2}{j-i+1}\right] = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} E\left[\frac{2}{k+1}\right] = O(n \log n)$$

Попытаемся решить задачу с помощью рекурентной формулы. За rank(x) обозначим количесво элементов в текущем массиве, не больших x.(x - случайно выбранный элемент.) Аналогично первому решению все значения rank(x) от 0 до n-1 равновероятны. При разбиении массива в левую часть попадет rank(x) - 1 элементов .Если T(n) - время работы алгоритма

$$T(n) = \frac{1}{n}(T(1) + T(n-1) + \sum_{q=1}^{n-1} (T(q) + T(n-q))) + \Theta(n) =$$

$$= \frac{1}{n} \sum_{q=1}^{n-1} (T(q) + T(n-q)) + \Theta(n) = \frac{2}{n} \sum_{q=1}^{n-1} T(k) + \Theta(n) = \Theta(n \log n)$$

при $T(1) = O(1)$, $T(n) = O(n^2)$

8.

Поймем, что partition займет $\Theta(i)$, если i-количество элементов. За базу возьмем n равное 1.

Если мы предположим по индукции, что мы умеем получать к-ую порядковую статистику для длины входа i, меньшей n за O(i). Тогда, исходя из рекурентной формулы мы сможем ее получить и для длины входа, равной n.

$$T(n) \leqslant \frac{1}{n} \left(\sum_{q=1}^{n} (T(q) + T(n-q)) \right) + \Theta(n) = O(n)$$

Следовательно сложность алгоритма O(n).