

Семестровая контрольная I

Необоснованные ответы не оцениваются! Если в задаче требуется построение алгоритма, то нужно построить оптимальный алгоритм (за неэффективность снижается оценка), доказать его корректность и оценить время работы (если не в условии не оговорено иное).

1 (3). Дан массив a вещественных чисел, отсортированный по возрастанию, а также числа p, q, r . Предложите алгоритм, строящий массив b , состоящий из чисел $px^2 + qx + r$, где x пробегает все элементы массива a , также отсортированный по возрастанию. Ограничение по времени — $O(n)$, по дополнительной памяти — $O(n)$. Считайте, что арифметические операции (и операции сравнения) с вещественными числами стоят $O(1)$.

2 (2). Верно ли для функций $f, g : \mathbb{N}_1 \rightarrow \mathbb{R}_{>0}$, что $f(n) = O(g(n))$ тогда и только тогда, когда выполняется условие: $\exists C > 0 : \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq C$.

3 (3). Найдите асимптотику роста функций, полагая, что $T(n) = \Theta(1)$ при малых n :

а) $T(n) = 25T(\frac{n}{4}) + n^3$; **б)** $T(n) = 243T(\frac{n}{3}) + 2n^5(\frac{\sqrt{n} + \log n}{\sqrt{n}})$.

4 (3+2+3). В оперативной памяти лежит массив длины n из натуральных (неотрицательных целых) чисел.

1. Постройте линейный алгоритм, который выводит наименьшее натуральное число, отсутствующее в массиве.

2. Решите задачу при условии, что массив отсортирован по неубыванию.

3. Докажите оптимальность вашего решения из предыдущего пункта.

5 (3). Планировщик работает следующим образом. В начале работы у него есть n задач с приоритетами p_1, p_2, \dots, p_n и временем исполнения каждой задачи t_1, t_2, \dots, t_n . На каждом шаге планировщик исполняет ровно одну задачу с максимальным приоритетом p_i , после чего приоритет всех задач в очереди увеличивается на время ожидания (на t_i или меньше для вновь добавленных задач). В планировщик могут добавляться и новые задачи — пары вида (p, t) пока исполняется какая-то задача, в момент добавления задачи известно время, которое отработала текущая выполняемая задача. Приоритеты и времена исполнений задач, уже поступивших в планировщик, меняться не могут (кроме как выше описанным образом). Постройте алгоритм, реализующий работу планировщика (алгоритм должен исполнять задачи в том же порядке, что и описанный планировщик), сложностью $O(m \log m)$ за всё время работы. Здесь m — общее число запросов, включая n исходных задач. В случае совпадения приоритетов у двух задач, первой исполняется задача с меньшим временем исполнения.

6 (4). Дан массив положительных целых чисел a . Предложите алгоритм, находящий для каждого элемента a индекс ближайшего слева элемента, меньшего его хотя бы в два раза. Если такого элемента нет, то должно возвращаться значение None. Ограничение по времени $O(n \log n)$, по дополнительной памяти — $O(n)$.