

1. Верно ли, что **а)**  $n = O(n \log n)$ ? **б)**  $\exists \varepsilon > 0 : n \log n = \Omega(n^{1+\varepsilon})$ ?

2. Известно, что  $f(n) = O(n^2)$ ,  $g(n) = \Omega(1)$ ,  $g(n) = O(n)$ . Положим

$$h(n) = \frac{f(n)}{g(n)}.$$

1. Возможно ли, что **а)**  $h(n) = \Theta(n \log n)$ ; **б)**  $h(n) = \Theta(n^3)$  ?

2. Приведите наилучшие (из возможных) верхние и нижние оценки на функцию  $h(n)$  и приведите пример функций  $f(n)$  и  $g(n)$  для которых ваши оценки на  $h(n)$  достигаются.

**3** [Шень 1.3.1 (а,б,г)]. Постройте линейный по времени онлайн-алгоритм, который вычисляет следующие функции или укажите индуктивные расширения для следующих функций:

**а)** среднее арифметическое последовательности чисел;

**б)** число элементов последовательности целых чисел, равных её максимальному элементу;

**в)** максимальное число идущих подряд одинаковых элементов;

**4.** Дано три отсортированных по возрастанию массива, внутри каждого массива все элементы различные. Предложите<sup>1</sup> линейный алгоритм нахождения числа различных элементов в объединении массивов.

**5.** Дана последовательность целых чисел  $a_1, a_2, \dots, a_n$ . Необходимо найти её самую длинную строго возрастающую подпоследовательность. Предложите **а)**  $O(n^2)$  алгоритм (докажите его корректность и асимптотику); **б\*)** [Шень 1.3.3]  $O(n \log n)$  алгоритм.

**6\*.** На вход подаётся последовательность натуральных чисел  $x_1, \dots, x_n$  в которой один из элементов встречается строго больше, чем  $\frac{n}{2}$  раз. Постройте алгоритм, который находит этот элемент, и при этом может использовать в качестве внешней памяти только стек (в который можно помещать только элементы последовательности), операции со стеком стоят  $O(1)$  времени; в оперативной памяти программа использует  $O(1)$  битов памяти и  $O(1)$  регистров (в каждом из которых может храниться число  $x_i$ ).

Числа  $x_i$  идут потоком данных на вход и каждое доступно для считывания только один раз — вернуться обратиться к прочитанным ранее числам можно, только если сохранить их в памяти.

---

<sup>1</sup>Здесь и всюду далее мы требуем не только описание алгоритма, но и доказательство его корректности, а также доказательство оценок на время работы алгоритма.