

1 (1+2). Верно ли, что **а)**  $n = O(n \log n)$ ? **б)**  $\exists \varepsilon > 0 : n \log n = \Omega(n^{1+\varepsilon})$ ?

2 (2+3). 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 (3). Дана программа

```
for (bound = 1; bound < n; bound *= 2) {
    for (i = 0; i < bound; i += 1) {
        for (j = 0; j < n; j += 2)
            печать ("алгоритм")
        for (j = 1; j < n; j *= 2)
            печать ("алгоритм")
    }
}
```

Пусть  $g(n)$  обозначает число слов “алгоритм”, которые напечатает соответствующая программа. Найдите  $\Theta$ -асимптотику  $g(n)$ .

4 [ 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;

**Комментарий:** В книге А. Шеня «Программирование. Теоремы и задачи» приведены задачи с решениями. Если в задаче ДЗ указана ссылка на эту книгу, то не нужно техать решение этих задач. Их нужно решить самостоятельно и свериться с решением в книге.

5 (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.

6 (4). На вход подаётся последовательность чисел  $a_1, b_1, a_2, b_2, \dots, a_n, b_n$ . Постройте онлайн-алгоритм, который вычисляет сумму  $\sum_{i \neq j} a_i \times b_j$ .

<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.

**7 (3).** 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 1.3.4]  $O(n \log n)$  алгоритм.

**8\*.** 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.