

## Домашнее задание

1. Алгоритм<sup>1</sup> получает на вход число  $n$  (в десятичной записи) и создаёт массив  $A[2, \dots, n]$ , заполненный нулями. Далее алгоритм выполняет следующую процедуру, пока массив не окажется заполнен единицами. Идёт по массиву от 2 до  $n$  пока не встретит первый ноль. Пусть ноль оказался в ячейке с номером  $k$ . Тогда алгоритм выводит  $k$  и заполняет все ячейки с номерами, кратными  $k$ , единицами: идёт по массиву дальше с шагом один и через каждые  $k$  клеток записывает в ячейку единицу.

1. Какую последовательность чисел выводит алгоритм?
  2. Оцените временную сложность алгоритма.
  3. Является ли алгоритм полиномиальным?
2. Верно ли, что **а)**  $n = O(n \log n)$ ? **б)**  $\exists \varepsilon > 0 : n \log n = \Omega(n^{1+\varepsilon})$ ?
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.

4. Найдите  $\Theta$ -асимптотику  $\sum_{i=1}^n \sqrt{i^3 + 2i + 5}$ .

5. Пусть для положительной функции  $f(n)$  известно, что  $f(n) = (3 + o(1))^n + \Theta(n^{100})$ . Верно ли в общем случае, что  $\log f(n) = \Theta(n)$ ?

6. Дана программа

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

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

7\*. Оцените асимптотику роста функции  $f(n) = \binom{2n}{n}$ .

---

<sup>1</sup>Алгоритм в этой задаче можно оптимизировать, но это не должно вас смущать. В нашем курсе (как и стандартно в математических курсах) нужно решать сформулированную задачу, поэтому, пожалуйста, удержитесь от порывов оптимизировать алгоритм и оценивать его сложность, если этого не требуется в условии.