Во всех домашних заданиях можно основываться на тех фактах, которые мы доказывали с вами на лекциях, все остальные факты доказывать. Исключения составляют достаточно известные и в то же время сложнодоказуемые математические факты, которые иногда могут приниматься на веру. Если вы не знаете, нужно ли какое-то конкретное утверждение (не имеющее отношение к нашей теме) доказывать, то задумайтесь о решении без этого факта. В большинстве заданий это можно сделать.

1 Неделя 1

1.1 Письменная часть

1. Докажите по определению: $\sum\limits_{i=1}^{n+5} 2^i = \mathcal{O}(2^n)$
2. Докажите по определению: $\frac{n^3}{6} - 7n^2 = \Omega(n^3)$
3. Докажите при помощи определения Θ, что $\max(f(n), g(n)) = \Theta(f(n) + g(n))$, если $f(n)$ и $g(n)$ неотрицательны.
4. Упорядочите функции так, чтобы, если $f(n)$ стоит раньше $g(n)$, то $f(n) = \mathcal{O}(g(n))$: $(\sqrt{2})^{\log n}$, n^2, $n!$, $(\log n)!$, $(\frac{3}{2})^2$, n^3, $\log^2 n$, $\log n!$, $\log\log n$, 2^{2^n}, $n^{\frac{1}{\log n}}$, 1, n, $(n+1)!$, $2^{2^{n+1}}$, e^n, $n\log n$, $(\log n)^{\log n}$, $2^{\log n}$, $\sqrt{\log n}$, $n^{\log\log n}$, $n \cdot 2^n$, $4^{\log n}$.

1.2 Устная часть

1. Докажите или опровергните, все функции положительные:
   (a) Из $f(n) = \mathcal{O}(g(n))$ вытекает, что $g(n) = \mathcal{O}(f(n))$.
   (b) Из $f(n) = \mathcal{O}(g(n))$ вытекает, что $2^{f(n)} = \mathcal{O}(2^{g(n)})$.
   (c) $f(n) = \mathcal{O}((f(n)^2)$.
   (d) $f(n) = \Theta(f(\frac{n}{2}))$.
   (e) $f(n) = \mathcal{O}(g(n))$ вытекает, что $g(n) = \Omega(f(n))$.
2. Пусть $p(n) = \sum\limits_{i=0}^{d} a_i n^i$, где $a_d > 0$, полином степени d от n. Используя определения асимптотических обозначений, считая, что k константа, докажите, что:
   (a) Если $k \geq d$, то $p(n) = \mathcal{O}(n^k)$.
   (b) Если $k = d$, то $p(n) = \Theta(n^k)$.
   (c) Если $k \leq d$, то $p(n) = \Omega(n^k)$.
3. Придумайте пару положительных функций $f(n)$ и $g(n)$, что не выполнено ни одно из двух: $f(n) = \mathcal{O}(g(n))$ и $f(n) = \Omega(g(n))$.
4. Докажите или опровергните, что $\log(n!) = \Theta(n\log n)$.

5. Докажите или опровергните, что из $k \log k = \Theta(n)$ следует, что $k = \Theta\left(\frac{n}{\log n}\right)$.
6. Решите $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$, в этом и в следующих четырех заданиях докажите точную верхнюю и нижнюю оценку с помощью метода математической индукции.
7. Решите $T(n) = 4T(\frac{n}{2}) + n \log n$.
8. Решите $T(n) = 7T(\frac{n}{2}) + n^2$.
9. Решите $T(n) = 3T(\sqrt{n}) + \log n$.
10. Решите $T(n) = 2T(\frac{n}{2}) + \frac{n}{\log n}$.
11. Пусть у нас есть два целых числа, записанных в десятичной системе счисления: x и y. Пусть $x = a \cdot 10^k + b$ и $y = c \cdot 10^k + d$. Рассмотрим рекурсивный метод умножения этих чисел:

```
fun multiply(x, y):
  k = max(len(x), len(y)) / 2
  a = shiftRight(x, k)
  b = getLowest(x, k)
  c = shiftRight(y, k)
  d = getLowest(y, k)
  ac = multiply(a, c)
  bd = multiply(b, d)
  ad = multiply(a, d)
  cb = multiply(c, b)
  e = add(ad, cb)
  ans = shiftLeft(ac, 2 * k)
  ans = add(ans, bd)
  ans = add(ans, shiftLeft(e, k))
  return ans
```

Заметим следующий факт:

$$(a \cdot 10^k + b) \cdot (c \cdot 10^k + d) = a \cdot c \cdot 10^{2k} + b \cdot d + (a \cdot d + b \cdot c) \cdot 10^k,$$

а так как $a \cdot d + c \cdot b = (a + b) \cdot (c + d) - a \cdot c - b \cdot d$, то

$$(a \cdot 10^k + b) \cdot (c \cdot 10^k + d) = a \cdot c \cdot 10^{2k} + b \cdot d + ((a + b) \cdot (c + d) - a \cdot c - b \cdot d) \cdot 10^k.$$

Рассмотрим второй метод умножения, основанный на этом факте:

```
fun multiply2(x, y):
  k = max(len(x), len(y)) / 2
  a = shiftRight(x, k)
  b = getLowest(x, k)
```

```
    c = shiftRight(y, k)
    d = getLowest(y, k)
    ac = multiply2(a, c)
    bd = multiply2(b, d)
    e = multiply2(add(a, b), add(c, d))
    e = subtract(e, ac)
    e = subtract(e, bd)
    ans = shiftLeft(ac, 2 * k)
    ans = add(ans, bd)
    ans = add(ans, shiftLeft(e, k))
    return ans
```

В нашем коде `ac` $= a \cdot c$, `bd` $= b \cdot d$, `e` $= a \cdot d + c \cdot b = (a + b) \cdot (c + d) - a \cdot c - b \cdot d$. Функции `len`, `shiftRight`, `shiftLeft`, `getLowest`, `add`, `subtract` работают за линейное от длины аргумента и длины результата время. Оцените время работы обоих алгоритмов, если считать, что его вызывают от двух чисел длины n.