AI Masters, Алгоритмы-2025, осень, семестровая контрольная (midterm)

1[1] Функции $f_i(n)$ таковы, что $\exists C_1, C_2 > 0, N \in \mathbb{N} : \forall n > N \hookrightarrow C_1 n^{-1} \leq f_i(n) \leq C_2 n^1$
Найдите наилучшие оценки сверху и снизу на функцию $g_j(n) = \sum\limits_{i \neq j}^{n} f_i(n) f_j(n)$

2[1] Оцените сложность работы алгоритма.

```
def f(n):
    for i = 0; i < n**2; i += 1:
        for j = 0; j < n; j += 1:
            for m = 1; m < j**2; m *= 2:
                print("hehe")

            for m = 0; m < n; m += 1:
                print("hehe")
```

3[1] Докажите нижнюю оценку в $\Omega(n^2)$ на умножение двух квадратных матриц $n \times n$.

4[2] Обобщите алгоритм Карацубы на случай деления чисел на три части вместо двух. Оцените асимптотику получившегося алгоритма. Это будет быстрее, чем обычный алгоритм Карацубы?

5[2 = 0.4+0.4+0.4+0.4+0.4]

1. $T(n) = T(\frac{n}{6}) + T(\frac{n}{2}) + T(\frac{n}{3}) + n$
2. $T(n) = T(n-1) + n^{1,5}$
3. $T(n) = 125T(\frac{n}{25}) + n\sqrt{n}$
4. $T(n) = 27T(\frac{n}{3}) + n^4$
5. $T(n) = 27T(\frac{n}{3}) + n^2$

6[3] Докажите специальный случай псевдо-мастер-теоремы, найдя Θ-оценку для рекурренты
$T(n) = n^{\gamma} T(\frac{n}{a}) + n^k$, где $\gamma > \frac{1}{2}$, $k > 0$, $a \geq 2$.
Обратите внимание, что $f(n) = \Theta(g(n)) \not\Rightarrow 2^{f(n)} = \Theta(2^{g(n)})$

7[1] Верно ли, что $5^{123} \equiv 3 \mod 13$?

8[1] Профессор О.П. Рометчивый предложил Дурацкую сортировку (Silly sort). Это рекурсивная процедура, которая получает на вход массив, вызывает на первом подмассиве длины $\sqrt{n}$ сортировку пузырьком, на втором подмассиве длиной $\frac{n}{10}$ сортировку слиянием, а на третьем вызывается рекурсивно. Найдите время работы Дурацкой сортировки.

9[2] Оцените в битовой модели сложность вычисления квадратного корня из целого числа. Квадратный корень - самое большое целое число, квадрат которого не больше исходного числа.

10[1] Произошло поразительное: П позволено, прочее предосудительно. Пожалуйста, покажите последовательность преобразований, представляющих $13^{-2} \mod$ *пятнадцать* попроще. Просим поискать подобное покороче.

11[1] У Ягами Лайта есть специальная тетрадочка. При записи в неё какого-либо числа L это число становится 'сложным'. 'Сложное' число можно сложить с любым другим и написать вместо них остаток от деления второго числа на 'сложное' число и само сложное число. При сложении L с L остается одна копия L. При сложении двух отличных от L чисел остается их сумма. Предложите алгоритм, определяющий, можно ли сделать так, что в результате останется только L, для поступающего на вход массива и выбранного в нем числа L.