Задача 1:

Доказать: $\frac{n^3}{6} - 7n^2 = \Omega(n^3)$

Воспользуемся определением:

$\exists\, c \in \mathbb{R},\ c > 0,\ N \in \mathbb{N} : \forall n \in \mathbb{N} : n > N,\ \frac{n^3}{6} - 7n^2 > cn^3$

Возьмем $c = \frac{1}{12}$, разделим выражение на n^2 и умножим на 12.

$n > 84$

Т.е. при $n > 84$ это будет выполняться. Тогда возьмем $N = 84$.

Формула будет справедлива при таких $c, N \Rightarrow \frac{n^3}{6} - 7n^2 = \Omega(n^3)$

Задача 2:

$1,\ n^{\frac{1}{\log n}},\ (\frac{3}{2})^2,\ \log\log n,\ \sqrt{\log n},\ \log^2 n,\ (\sqrt{2})^{\log n},\ n,\ 2^{\log n},\ \log n!,\ n\log n,\ n^2$
$4^{\log n},\ n^3,\ (\log n)!,\ n^{\log\log n},\ (\log n)^{\log n},\ n \cdot 2^n,\ e^n,\ n!,\ (n+1)!,\ 2^{2^n},\ 2^{2^{n+1}}$

Задача 3:

Очевидно что $\log n! = \mathcal{O}(n \log n)$

$\log n! = \log 1 + \cdots + \log n \leqslant n \log n$

Докажем что $\log n! = \Omega(n \log n)$

$\log 1 + \cdots + \log \frac{n}{2} + \log(\frac{n}{2} + 1) + \cdots + \log n > \frac{n}{2} \log \frac{n}{2}$

$\frac{n}{2} \log \frac{n}{2} > cn \log n$

Пусть: $c = \frac{1}{4}$

Тогда сократим на n и отбросим возрастающий log

$\frac{n}{2} > \sqrt{n}$

Это верно при $n > 4 \Rightarrow \log n! = \Omega(n \log n) \Rightarrow \log n! = \Theta(n \log n)$

Задача 4:

Построим дерево рекурсии и посчитаем сумму операций.

Поймем что сумма равна $n \log\log n$

Докажем по индукции

База индукции очевидна

$T(c) = \mathcal{O}(1)$

Предположение

$T(n) = cn \log\log n = \mathcal{O}(n \log\log n)$

Переход

$T(n) = cn \log\log \frac{n}{2} + \frac{n}{\log n} < cn \log\log n$

$c \log n(\log\log n - \log\log \frac{n}{2}) > 1$

$c \log n(\log \frac{\log n}{\log \frac{n}{2}}) > 1$

$\log\,(\frac{\log n}{\log \frac{n}{2}})^{c \log n} > 1$

$(\frac{\log n}{\log n - 1})^{c \log n} > 2$

Сделаем замену: $k = \log n - 1$

Пусть: $c = 1$

$f_1(k) = (1 + \frac{1}{k})^{k+1} > 2$

$\lim_{k \to \infty} f_1(k) = e$

Причем $\forall k > 2, f_1(k) > e$ так как функция убывает (известный факт)
$(1+\frac{1}{k})^{k+1} > 2 \Rightarrow T(n) = \mathcal{O}(n \log\log n)$
Теперь докажем что $T(n) = \Omega(n \log\log n)$
Аналогичные рассуждения индукции
Возьмем $c = \frac{1}{8}$
$f_2(k) = (1+\frac{1}{k})^{\frac{k+1}{8}} < 2$
$\lim_{k\to\infty} f_2(k) = \sqrt[8]{e}$
Причем $\forall k > 2, f_1(k) > \sqrt[8]{e} \Rightarrow \exists k_1 : f_2(k_1) < \sqrt[8]{e} < 2 \Rightarrow$
$\Rightarrow T(n) = \Omega(n \log\log n) \Rightarrow T(n) = \Theta(n \log\log n)$

Задача 5:
Рассмотрим первый алгоритм:
На каждом шаге оба аргуманта делятся на 2 так как в качестве k выбирается $\frac{n}{2}$. Значит $\max(n_1, n_2)$ каждый раз будет уменьшаться вдвое. Каждый вызов функции делает $\mathcal{O}(n)$ операций и рекурсивно вызывается 4 раза. Тогда можно записать формулу количества операций.
$T_1(n) = 4 \cdot T_1(\frac{n}{2}) + c \cdot n$
Константу можно вынести.
Применим Мастер теорему к $T_1(n)$
Получим что $T_1(n) = cn^2 \Rightarrow T_1(n) = \Theta(n^2)$
Рассмотрим второй алгоритм:
Рассуждения аналогичны. Разница есть лишь в количестве вызовов рекурсии, их здесь 3.
$T_2(n) = 3 \cdot T_2(\frac{n}{2}) + c \cdot n$
Выносим константу и используем Мастер теорему.
$T_2(n) = cn^{\log_2 3} \Rightarrow T_2(n) = \Theta(n^{\log_2 3})$