Пункт 1.

```
algorithm1(A, n)
if n ≤ 20
    return A[n]
x = algorithm1(A, n - 5)
for i = 1 to ⌊n/2⌋
    for j = 1 to ⌊n/2⌋
        A[i] = A[i] - A[j]
x = x + algorithm1(A, n - 8)
return x
```

```
algorithm2(A, n)
  if n ≤ 50
      return A[n]
  x = algorithm2(A, ⌊n/4⌋)
  for i = 1 to ⌊n/3⌋
      A[i] = A[n - i] - A[i]
  x = x + algorithm2(A, ⌊n/4⌋)
  return x
```

$$T_{A_1}(n) = \begin{cases} \Theta(1), n \leq 20 \\ T_{A_1}(n-5) + \Theta\left(\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor\right) + T_{A_1}(n-8), \text{ иначе} \end{cases}$$

$$\Theta\left(\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor\right) = \Theta(n^2) \implies T_{A_1}(n) = \begin{cases} \Theta(1), n \leq 20 \\ T_{A_1}(n-5) + T_{A_1}(n-8) + \Theta(n^2), \text{ иначе} \end{cases}$$

$$T_{A_2}(n) = \begin{cases} \Theta(1), n \leq 50 \\ T_{A_2}(\left\lfloor \frac{n}{4} \right\rfloor) + \Theta\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + T_{A_2}(\left\lfloor \frac{n}{4} \right\rfloor), \text{ иначе} \end{cases}$$

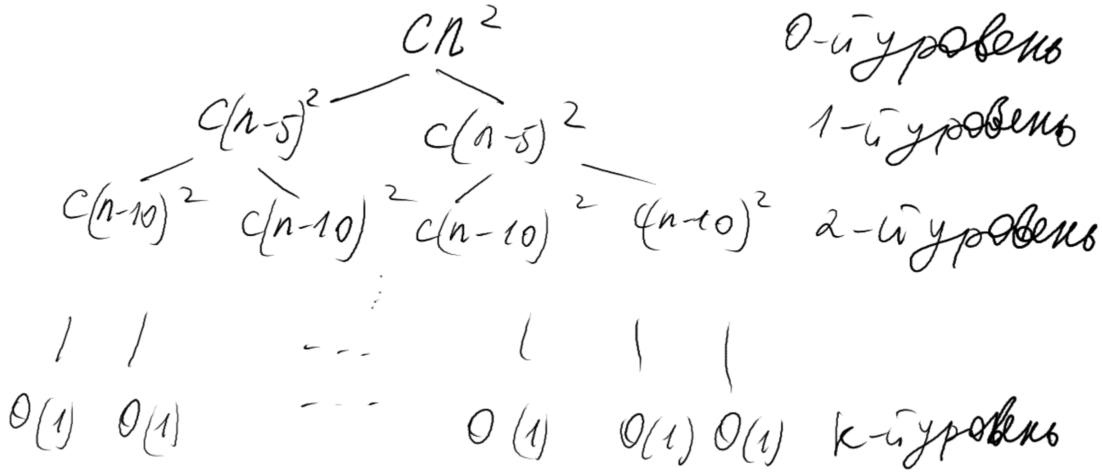
$$\Theta\left(\left\lfloor \frac{n}{3} \right\rfloor\right) = \Theta(n) \implies T_{A_2}(n) = \begin{cases} \Theta(1), n \leq 50 \\ 2T_{A_2}(\left\lfloor \frac{n}{4} \right\rfloor) + \Theta(n), \text{ иначе} \end{cases}$$

Ответ:
$$T_{A_1}(n) = \begin{cases} \Theta(1), n \leq 20 \\ T_{A_1}(n-5) + T_{A_1}(n-8) + \Theta(n^2), \text{ иначе} \end{cases}$$
$$T_{A_2}(n) = \begin{cases} \Theta(1), n \leq 50 \\ 2T_{A_2}(\left\lfloor \frac{n}{4} \right\rfloor) + \Theta(n), \text{ иначе} \end{cases}$$

Пункт 2.

Алгоритм 1

При $n > 20 : T_{A_1}(n) = T_{A_1}(n-5) + T_{A_1}(n-8) + \Theta(n^2)$
Попробуем найти оценку для $T_3(n) = T_3(n-5) + T_3(n-5) + \Theta(n^2)$, где $T_3(n) = \Theta(1)$ при $n \leq 4$, и предположим, что она верна и для T_{A_1}, доказав методом подстановки.
Построим дерево рукурсии для T_3:



где $k = \left\lfloor \frac{n}{5} \right\rfloor$

На i-ом уровне выполняется порядка $c(n-5i)^2 * 2^i$ операций
(включая первый и последний уровни при i = 0 и i = k соответственно)
$\implies T_3(n) = \sum_{i=0}^{k} c(n-5i)^2 2^i$

$$\sum_{i=0}^{k} c(n-5i)^2 2^i = c\sum_{i=0}^{k}\left(n^2 2^i - 10ni2^i + 25i^2 2^i\right)$$

Вычислим $\sum\limits_{i=0}^{k} n^2 2^i$:

$$\sum_{i=0}^{k} n^2 2^i = n^2\sum_{i=0}^{k} 2^i = n^2\frac{2^{k+1}-1}{2-1} = n^2(2^{k+1}-1)$$

Вычислим $\sum\limits_{i=0}^{k} -10ni2^i$:

$$\sum_{i=0}^{k} -10ni2^i = -20n\sum_{i=0}^{k} i2^{i-1}$$

Обозначим функциональный ряд $S_n(x) = \sum\limits_{i=0}^{k} ix^{i-1}$

$$\sum_{i=0}^{k} ix^{i-1} = \sum_{i=0}^{k}\left(x^i\right)'_x = \left(\sum_{i=0}^{k} x^i\right)'_x = \left(\frac{x^{k+1}-1}{x-1}\right)'_x = \frac{(k+1)x^k(x-1)-(x^{k+1}-1)}{(x-1)^2}$$

$$\sum_{i=0}^{k} i2^{i-1} = S_n(2) = \frac{(k+1)2^k(2-1)-(2^{k+1}-1)}{(2-1)^2} = (k-1)2^k+1 \implies$$

$$\implies \sum_{i=0}^{k} -10ni2^i = -20n((k-1)2^k+1)$$

Вычислим $\sum\limits_{i=0}^{k} 25i^2 2^i$:

$$\sum_{i=0}^{k} 25i^2 2^i = 25\sum_{i=0}^{k}\left(i(i-1)2^i + i2^i\right) = 25\sum_{i=0}^{k} i(i-1)2^i + 25\sum_{i=0}^{k} i2^i =$$

$$= 100\sum_{i=0}^{k} i(i-1)2^{i-2} + 50\sum_{i=0}^{k} i2^{i-1} = 100\sum_{i=0}^{k} i(i-1)2^{i-2} + 50((k-1)2^k+1)$$

Обозначим функциональный ряд $S_n(x) = \sum\limits_{i=0}^{k} i(i-1)x^{i-2}$

$$\sum_{i=0}^{k} i(i-1)x^{i-2} = \left(\sum_{i=0}^{k} x^i\right)''_{xx} = \left(\frac{x^{k+1}-1}{x-1}\right)''_{xx} = \left(\frac{(k+1)x^k(x-1)-(x^{k+1}-1)}{(x-1)^2}\right)'_x =$$

$$= \left(\frac{kx^{k+1}-kx^k-x^k+1}{(x-1)^2}\right)'_x =$$

$$= \frac{(k(k+1)x^k-k^2x^{k-1}-kx^{k-1})(x-1)^2-(kx^{k+1}-kx^k-x^k+1)2(x-1)}{(x-1)^4} =$$

$$= \frac{k(k+1)(x^k-x^{k-1})(x-1)-2(kx^k(x-1)-x^k+1)}{(x-1)^3} =$$

$$= \frac{k(k+1)x^{k-1}(x-1)(x-1)-2(kx^k(x-1)-x^k+1)}{(x-1)^3} \implies$$

$$\implies \sum_{i=0}^{k} i(i-1)2^{i-2} = S_n(2) = \frac{k(k+1)2^{k-1}(2-1)(2-1)-2(k2^k(2-1)-2^k+1)}{(2-1)^3} =$$

$$= k(k+1)2^{k-1} - 2(k2^k-2^k+1)$$

Получим:

$$\sum_{i=0}^{k} 25i^2 2^i = 100(k(k+1)2^{k-1} - 2(k2^k - 2^k + 1)) + 50((k-1)2^k + 1) =$$
$$= 50(2(k(k+1)2^{k-1} - 2(k2^k - 2^k + 1)) + (k-1)2^k + 1) =$$
$$= 50(2(k^2 2^{k-1} + k2^{k-1} - k2^{k+1} + 2^{k+1} - 2) + (k-1)2^k + 1) =$$
$$= 50(k^2 2^k + k2^k - k2^{k+2} + 2^{k+2} - 4 + k2^k - 2^k + 1) =$$
$$= 50(k^2 2^k - k2^{k+1} + 3 * 2^k - 3)$$

Итого:

$$T_3(n) = c(n^2(2^{k+1} - 1) - 20n((k-1)2^k + 1) + 50(k^2 2^k - k2^{k+1} + 3 * 2^k - 3)) =$$
$$= c(n^2 2^{k+1} - n^2 - 20nk2^k + 20n2^k - 20n + 50k^2 2^k - 50k2^{k+1} + 150 * 2^k - 150) =$$
$$= c(2n^2 2^k - 20nk2^k - n^2 + 20n2^k - 20n + 50k^2 2^k - 100k2^k + 150 * 2^k - 150) =$$
$$= c(2n^2 2^k - 20nk2^k + 50k^2 2^k + 20n2^k - 100k2^k - n^2 - 20n + 150 * 2^k - 150) =$$
$$= c(2^{k+1}(n-5k)^2 + 20 * 2^k(n-5k) + 150 * 2^k - n^2 - 20n - 150)$$
$$k = \left\lfloor \frac{n}{5} \right\rfloor \implies 0 \le n - 5k \le 4 \implies$$
$$\implies c(150 * 2^k - n^2 - 20n - 150) \le T_3(n) \le$$
$$\le c(2^{k+1} * 16 + 20 * 2^k * 4 + 150 * 2^k - n^2 - 20n - 150) \implies$$
$$\implies c(150 * 2^k - n^2 - 20n - 150) \le T_3(n) \le c(262 * 2^k - n^2 - 20n - 150) \implies$$
$$\implies T_3(n) = \Theta(2^k) = \Theta(2^n), \text{ т.к. } k = \left\lfloor \frac{n}{5} \right\rfloor$$

Т.к. при $n > 20 : T_{A_1}(n) = T_{A_1}(n-5) + T_{A_1}(n-8) + \Theta(n^2)$, а при $n > 4 :$
$T_3(n) = T_3(n-5) + T_3(n-5) + \Theta(n^2)$ и $T_3(n) = \Theta(2^n)$,
то предположим, что и $T_{A_1}(n) = \Theta(2^n)$
Докажем это методом подстановки:

a. $T_{A_1}(n) = \mathcal{O}(2^n)$

$T_{A_1}(n) = \mathcal{O}(2^n) \iff \exists c > 0 :$
$T_{A_1}(n) = T_{A_1}(n-5) + T_{A_1}(n-8) + c_1 n^2 = c2^{n-5} + c2^{n-8} + c_1 n^2 \le c2^n$
$c2^{n-5} + c2^{n-8} + c_1 n^2 \le c2^n \iff$
$\iff c_1 n^2 \le c2^n - (c2^{n-5} + c2^{n-8}) \iff$
$\iff c_1 n^2 \le \frac{247}{256} c2^n$ (1)
Положим $c := 2c_1$, тогда:
(1) $\iff n^2 \le \frac{247}{128} 2^n$ (2)
Из курса математического анализа известно, что $\forall n (n \in \mathbb{N} \land n \ge 4) \implies n^2 \le 2^n$
(доказательство методом математической индукции)
Следовательно, при $n \ge 4 : n^2 \le 2^n \implies$
$\implies$ при $n \ge 4$ выполняется (2) $\implies$
$\implies$ при $n \ge 4$ и $c = 2c_1$ выполняется (1) $\implies$
$\implies$ при $n \ge 4$ и $c = 2c_1 : T_{A_1}(n) = \mathcal{O}(2^n)$

b. $T_{A_1}(n) = \Omega(2^n)$

Доказательство аналогично пункту а., однако оценивание
$T_{A_1}(n-5) + T_{A_1}(n-8) + c_1 n^2$ снизу

Получили:

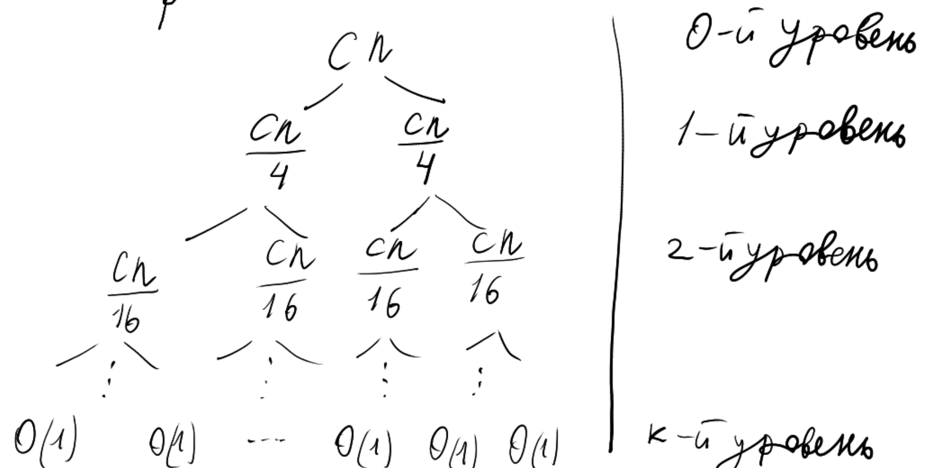
$(T_{A_1}(n) = \mathcal{O}(2^n)) \wedge (T_{A_1}(n) = \Omega(2^n)) \implies T_{A_1}(n) = \Theta(2^n)$

Алгоритм 2

$$T_{A_2}(n) = \begin{cases} \Theta(1), n \le 50 \\ 2T_{A_2}(\lfloor \frac{n}{4} \rfloor) + \Theta(n), \text{ иначе} \end{cases}$$

Найдём и докажем асимптотическую оценку функции временной сложности второго алгоритма при помощи дерева рекурсии

При $n > 50$



$K = \lceil \log_4 \frac{n}{50} \rceil$, т.к. $K = \min\{m \in \mathbb{N} \cup \{0\} \mid \frac{n}{4^m} \le 50\}$

На i-ом уровне выполняется $c\frac{n}{4^i}2^i$ операций
(включая первый и последний уровни при i = 0 и i = k соответственно)
$\implies T_{A_2}(n) = \sum_{i=0}^{k} c\frac{n}{4^i}2^i$

$$\sum_{i=0}^{k} c\frac{n}{4^i}2^i = cn\sum_{i=0}^{k}\frac{1}{2^i} = cn\frac{\left(\frac{1}{2}\right)^{k+1} - 1}{\frac{1}{2} - 1} = cn\left(2 - \frac{1}{2^k}\right) = cn\left(2 - \frac{1}{2^{\log_4 \frac{n}{50}}}\right) =$$

$$cn\left(2 - \frac{1}{\sqrt{2^{\log_2 \frac{n}{50}}}}\right) = cn\left(2 - \frac{1}{\sqrt{\frac{n}{50}}}\right) = cn\left(2 - \frac{\sqrt{50}}{\sqrt{n}}\right) = 2cn - c\sqrt{50n} \implies$$

$$\implies T_{A_2}(n) = \Theta(n)$$

Ответ: $T_{A_1}(n) = \Theta(2^n)$; $T_{A_2}(n) = \Theta(n)$