A1. Временная сложность рекурсии

Ниже приведены два рекурсивных алгоритма обработки целочисленного массива ***A*** размера ***n***.

```
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
```

1. Для каждого из представленных алгоритмов составить рекуррентное соотношение, которое выражает их временную сложность $T(n)$. Обратите внимание, что рекуррентное соотношение должно давать полное представление о сложности алгоритма, т.е., охватывать как рекурсивную, так и нерекурсивную ветку вычислений. Предполагается, что все арифметические операции выполняются за постоянное время.

алгоритм 1:

$$\begin{cases} T(N) = T(N-5) + T(N-8) + \mathrm{c}N^2 \\ T(K) = \mathrm{c}, k \leq 20 \end{cases}$$

алгоритм 2:

$$\begin{cases} T(N) = 2T(N/4) + \mathrm{c}N \\ T(K) = \mathrm{c}, K \leq 50 \end{cases}$$

2. Вычислите асимптотическую точную границу $\Theta(f(n))$ временной сложности для каждого из представленных алгоритмов, если это возможно. В случае невозможности формирования асимптотической точной границы, представить отдельно верхнюю и нижнюю границы. Обоснуйте свой ответ с помощью метода подстановки, дерева рекурсии, или итерации.

алгоритм1:

заметим что

$$T_1(n) \leq T(n) \leq T_2(n)$$

где

$$T_1(n) = 2T(n-8) + cn^2$$

$$T_2(n) = 2T(n-5) + cn^2$$

найдем верхнюю границу для $T_2(n)$

воспользуемся методом подстановки:

допустим верхняя граница $O(2^{n/5}n^2)$

$2T(n-5)+c_1n^2 < c_2 2^{n/5}n^2$

$2T(n-5)+c_1n^2 < 2(c_2 2^{n/5-1}(n-5)^2)+c_1n^2 < c_2 2^{n/5}n^2$

$c_2 2^{n/5}(n-5)^2+c_1n^2 < c_2 2^{n/5}n^2$

$c_1n^2+25c_2 2^{n/5} < 10c_2 2^{n/5}n$

$O(2^{n/5}) < O(2^{n/5}n^2)$

получается для $T_2(n)$ тогда и для $T(n)$ верхняя граница будет: $O(2^{n/5}n^2)$

найдем нижнюю границу для $T_1(n)$

также воспользуемся методом подстановки, допустим нижняя граница $c_2 2^{n/9}n^2$

$2T(n-8)+c_1n^2 < c_2 2^{n/9}n^2$

$c_2 2(2^{\frac{n-8}{9}}(n-8)^2)+c_1n^2 < c_2 2^{n/9}n^2$

$c_2 2^{\frac{n+1}{9}}n^2 - 16c_2 2^{\frac{n+1}{9}}n^2 + 64c_2 2^{\frac{n+1}{9}} + c_1n^2 < c_2 2^{n/9}n^2$

$64c_2 2^{\frac{n+1}{9}} + c_1n^2 < 16c_2 2^{\frac{n+1}{9}}n^2 + 2^{n/9}n^2(c_1 - c_2\sqrt[9]{2})$

чтобы неравенство выполнялось, надо чтобы $c_1 > c_2\sqrt[9]{2}$

например при $c_2 = \frac{c_1}{\sqrt[9]{2}} - 1$ условие выполнится

тогда $T_1(n) = \Omega(2^{n/9}n^2)$, тогда и $T(n) = \Omega(2^{n/9}n^2)$

нашли обе границы:

$T(n) = O(2^{n/5}n^2)$

$T(n) = \Omega(2^{n/9}n^2)$

алгоритм2:

воспользуемся методом итерации:

$T(N) = 2T(N/4) + cN = 4T(N/16) + \frac{cN}{2} + cN = 2^k(T/4^k) + \sum_{i=0}^{k} \frac{cN}{2^i} = \sum_{i=0}^{log_4(N-50)} \frac{cN}{2^i} = cN\sum\frac{1}{2^i} = 2cN$

получилось:

$T(N) = c'N$

тогда $T(N) = \Theta(N)$

покажем:

$c‘N \leq C_1N$

для любого N при $C_1 = c' + 1$

$c'N \geq C_2 N$

для любого N при $C_2 = c' - 1$