1 (). К серверу приходят одновременно n клиентов. Для клиента i известно время его обслуживания t_i. Время ожидания клиента определяется как сумма времени обслуживания всех предыдущих клиентов и времени обслуживания его самого. К примеру, если обслуживает клиентов в порядке номеров, то время ожидания клиента i будет равно $\sum_{j=1}^{i} t_j$. Постройте эффективный алгоритм, находящий последовательность обслуживания клиентов с минимальным суммарным временем ожидания клиентов, докажите его корректность и оцените асимптотику.

▷ Заметим, что если у i-го подошедшего клиента время облуживания t_i, то суммарное вермя ожидания будет

$$T = \sum_{i=1}^{n}(n-i+1)t_i$$

Далее рассмотрим $t_i, t_j : i < j, t_i \geqslant t_j$. Если мы поменяем этих клинтов местами, T уменьшится на $(j-i)t_i$ и увеличится на $(j-i)t_j$, в результате T уменьшится на $(j-i)(t_i-t_j) \geqslant 0$, т. е. T не увеличится. Получаем, что расстановка с наименьшим суммарным временем ожидания — это расстановка при которой время ожидания каждого следующего клиента не убывает. Алгоритм тем самым должен просто отсортировать клиентов по убыванию времени ожидания. Для этого подойдет сортировка слиянием, работающая за $O(n \log n)$, что является наилучшим вариантом с точки зрения ассимптотики. ◁

2 (). Найдите асимптотическую оценку функции $T(n)$. Примените мастер-теорему в тех случаях, когда ее можно использовать.

1. $T(n) = 25T(\frac{n}{5}) + n^2$
2. $T(n) = 16T(\frac{n}{2}) + n^3$
3. $T(n) = 9T(\frac{n}{3}) + n^3$
4. $T(n) = T(n-1) + 3n$
5. $T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + n$

▷ Если $T(n) = aT(\frac{n}{b}) + O(n^c)$, то

- $T(n) = \Theta(n^{\log_b a})$, $c < \log_b a$
- $T(n) = \Theta(n^c \log n)$, $c = \log_b a$
- $T(n) = \Theta(n^c)$, $c > \log_b a$

1. $\log_b a = 2, c = 2 \Rightarrow T(n) = \Theta(n^2 \log n)$
2. $\log_b a = 4, c = 3 \Rightarrow T(n) = \Theta(n^4)$
3. $\log_b a = 2, c = 3 \Rightarrow T(n) = \Theta(n^3)$
4. $T(0) = T_0, T(n) = T_0 + \sum\limits_{i=1}^{n} 3i = T_0 + 3\frac{n(n+1)}{2} = \Theta(n^2)$

◁

3 (). Оцените асимптотически, сколько раз будет напечатана строка "heh" при вызове функции f.

Ответ: $\Theta(n^{\log_2 3})$.

▷ За $T(n)$ обозначим кол-во "heh" при вызове $f(n)$. Тогда уз условия получаем:

$$T(n) = 3T(\frac{n}{2}) + \frac{n}{2}.$$

$\log_b a = \log_2 3 > 1 = c$, откуда по мастер-теореме

$$T(n) = \Theta(n^{\log_2 3}).$$

◁