1 Асимптотика и линейные алгоритмы

1.1 Практика

Напомним определения:

- $f(n) \in \mathcal{O}(g(n)) \equiv \exists N, C > 0 : \forall n \geq N : f(n) \leq C \cdot g(n)$
- $f(n) \in \Omega(g(n)) \equiv \exists N, C > 0 : \forall n \geq N : C \cdot g(n) \leq f(n)$
- $f(n) \in \Theta(g(n)) \equiv \exists N, C_1 > 0, C_2 > 0 : \forall n \geq N : C_1 \cdot g(n) \leq f(n) \leq C_2 \cdot g(n)$
- $f(n) \in o(g(n)) \equiv \forall C > 0 : \exists N : \forall n \geq N : f(n) < C \cdot g(n)$
- $f(n) \in \omega(g(n)) \equiv \forall C > 0 : \exists N : \forall n \geq N : C \cdot g(n) < f(n)$

Все функции здесь $\mathbb{N} \to \mathbb{N}$ или $\mathbb{N} \to \mathbb{R}_{>0}$ (далее будет ясно из контекста, какой класс функций используется). В дальнейшем, когда речь идет о принадлежности функций вышеопределенным множествам, мы будем использовать знак “=” вместо “∈”, т.к. в литературе обычно используются именно такие обозначения.

Асимптотики

1. Докажите, что:
   (a) $f(n) = \Omega(g(n)) \Leftrightarrow g(n) = \mathcal{O}(f(n))$
   (b) $f(n) = \omega(g(n)) \Leftrightarrow g(n) = o(f(n))$
   (c) $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = \mathcal{O}(g(n)) \wedge f(n) = \Omega(g(n))$

2. **Контекст имеет значение**
   Правда ли, что $f(n) = \mathcal{O}(f(n)^2)$?

3. **Классы**
   Определим отношение “$\sim$”. Будем говорить, что $f \sim g$, если $f = \Theta(g)$. Покажите, что $\sim$ — отношение эквивалентности, т.е. оно
   - Рефлексивное: $\forall f : f \sim f$,
   - Симметричное: $\forall f, g : f \sim g \Leftrightarrow g \sim f$,
   - Транзитивное: $\forall f, g, h : (f \sim g) \wedge (g \sim h) \Rightarrow f \sim h$.

4. **Порядки**
   Определим отношение “$\preceq$”. Будем говорить, что $f \preceq g$, если $f = \mathcal{O}(g)$.
   Определим отношение $f \preceq g \equiv f = \mathcal{O}(g)$.
   (a) Докажите, что $\preceq$ — отношение предпорядка (рефлексивное и транзитивное)
   (b) Докажите, что $\preceq$ — не отношение частичного порядка, так как не удовлетворяет антисимметричности
   (c) Докажите, что $\preceq$ — отношение частичного порядка на классах эквивалентности по $\sim$?

5. Считайте, что функции здесь $\mathbb{N} \to \mathbb{N}$ и $\forall n : f(n) > 1 \wedge g(n) > 1$.
   (a) $f(n) = \Omega(f(n/2))$?
   (b) $f(n) = \mathcal{O}(g(n)) \Rightarrow \log f(n) = \mathcal{O}(\log g(n))$?
   (c) $f(n) = \mathcal{O}(g(n)) \Rightarrow 2^{f(n)} = \mathcal{O}(2^{g(n)})$?
   (d) $f(n) = o(g(n)) \Rightarrow \log f(n) = o(\log g(n))$?
   (e) $f(n) = o(g(n)) \Rightarrow 2^{f(n)} = o(2^{g(n)})$?
   (f) $\sum_{k=1}^{n} \frac{1}{k} = \Omega(\log n)$?

6. Определить асимптотику (считайте, что при $x \leq 100$ будет выполняться $T(x) = 100$).

   (a) $T(x) = T(a) + T(x - a) + n$ для натурального числа a.

   (b) $T(x) = T(\frac{x}{2}) + 1$.

   (c) $T(x) = 2 \cdot T(\sqrt{x}) + \log x$

Линейные алгоритмы

7. Вектор (расширяющийся массив) кроме операций `get(i)` и `set(i, x)` как у обычного массива умеет выполнять `push_back(x)` (`append(x)`) и `pop_back()` (`pop()`).

   Часто вектор реализуют следующим образом: храним обычный массив с некоторым запасом по памяти. Когда память заканчивается (из-за операций `push_back`), выделяем новый отрезок памяти большего размера, копируем туда текущий массив, старую память удаляем.

   Сравним две схемы перевыделения памяти:

   (a) При заполнении отрезка памяти размера n выделяем новый отрезок размером $n + c$ для какой-то константы c.

   (b) При заполнении отрезка памяти размера n выделяем новый отрезок размером cn для какой-то константы c.

   Оцените среднее (т. н. *амортизированное*) время работы операций для каждой из схем. На что влияет выбор константы c?

8. Дан массив целых чисел a_i. Придумайте структуру данных, которая бы умела отвечать на запросы вида “По данным l и r вернуть $\sum_{i=l}^{r} a_i$” за $\mathcal{O}(1)$.

   Разрешается сделать предподсчёт за $\mathcal{O}(n)$. Значения в массиве не меняются.

9. Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{N}$ и $S \in \mathbb{N}$. Найти l, r $(1 \leq l \leq r \leq n)$ такие, что сумма $\sum_{i=l}^{r} a_i = S$. Задачу требуется решить за линейное от n время.

10. Дана скобочная последовательность, составленная из скобок `'('`, `')'`, `'['`, `']'`, `'{'`, `'}'`. Последовательность называется корректной, если каждой открывающей скобке соответствует закрывающая скобка того же типа, и соблюдается вложенность. Примеры: `([{}])` и `()()` – корректные, а `[)` и `[(])` – нет.

    Придумайте алгоритм, который проверяет корректность последовательности за линейное время.

11. Пусть элементы здесь линейно упорядочены и мы умеем сравнивать их за $\mathcal{O}(1)$.

    (a) Придумайте стек, в котором можно узнавать минимум за $\mathcal{O}(1)$. Все остальные операции стека также должны работать за $\mathcal{O}(1)$.

    (b) Придумайте очередь, в которой можно узнавать минимум за $\mathcal{O}(1)$. Все остальные операции очереди должны работать за амортизированное $\mathcal{O}(1)$.

    (c) Придумайте вариант очереди с минимумом на основе пары из обычной очереди и дека потенциальных минимумов.

12. Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{N}$. Для каждого a_i найти самый правый из элементов, которые левее и не больше его. Задачу требуется решить за линейное от n время.

13. Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{N}$. Найти l, r $(1 \leq l \leq r \leq n)$ такие, что

    (a) значение $(r - l + 1) \min_{i \in [l,r]} a_i$ было бы максимально.

    (b) значение $\left(\sum_{i \in [l,r]} a_i\right) \min_{i \in [l,r]} a_i$ было бы максимально.

    Задачу требуется решить за линейное от n время.

14. Вам дан массив натуральных чисел и число k. Требуется найти подотрезок массива такой, что НОК чисел на нем равен k или заявить, что такого нет. Время работы: $\mathcal{O}(nT_{LCM}(k))$, где $T_{LCM}(k)$ — время подсчета НОК для чисел размера k.

1.2 Домашнее задание

1. Дайте ответ для двух случаев $\mathbb{N} \to \mathbb{N}$ и $\mathbb{N} \to \mathbb{R}_{>0}$:
   (a) Если в определении $\mathcal{O}$ опустить условие про N (т.е. оставить просто $\forall n$), будет ли полученное определение эквивалентно исходному?
   (b) Тот же вопрос про o.

2. Считайте, что функции здесь $\mathbb{N} \to \mathbb{N}$ и $\forall n : f(n) > 1 \wedge g(n) > 1$.
   (a) $f(n) = \mathcal{O}(g(n)) \Rightarrow \log f(n) = \mathcal{O}(\log g(n))$?
   (b) $f(n) = o(g(n)) \Rightarrow 2^{f(n)} = o(2^{g(n)})$?
   (c) $\sum_{k=1}^{n} \frac{1}{k} = \Omega(\log n)$?

3. Заполните табличку, поясните строчки 4 и 7:

A	B	$\mathcal{O}$	o	Θ	ω	Ω
n	n^2	$+$	$+$	$-$	$-$	$-$
$\log^k n$	n^ϵ					
n^k	c^n					
$\sqrt{n}$	$n^{\sin n}$					
2^n	$2^{n/2}$					
$n^{\log_2 m}$	$m^{\log_2 n}$					
$\log(n!)$	$\log(n^n)$					

   Здесь все буквы, кроме n, — положительные константы.

4. Вам дан массив из n элементов и список из m запросов $add(x, l, r)$: прибавить x к каждому элементу на отрезке $[l, r]$. За $\mathcal{O}(n + m)$ выведите массив, получающийся из исходного после выполнения заданных запросов.

   Подсказка: попробуйте решить задачу, где необходимо прибавлять число не на произвольном подотрезке, а на каком-то суффиксе, то есть, когда $r = n$.

5. `[*]`[1] Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{N}$.
   (a) Найти l, r $(1 \le l \le r \le n)$ такие, что значение $(r - l + 1) \min_{i \in [l,r]} a_i$ было бы максимально.
   (b) Найти l, r $(1 \le l \le r \le n)$ такие, что значение $\left(\sum_{i \in [l,r]} a_i\right) \min_{i \in [l,r]} a_i$ было бы максимально.

   Задачи требуется решить за линейное от n время.

[1] дополнительная задача со звёздочкой, чтобы набрать полный балл, её необязательно решать