Практика по алгоритмам, ВШЭ

Владислав Кораблинов, Антон Гардер*

Осень, 2020

*Составители сборника не всегда являются авторами задач. Авторы не указаны в учебных целях.

1 Практика 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 "$\in$", 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. 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.

8. 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 более эффективный по памяти вариант очереди с минимумом на основе пары из обычной очереди и дека.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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 здесь, что $\forall n : f(n) > 1 \wedge g(n) > 1$. Правда ли, что $f(n) = o(g(n)) \Rightarrow 2^{f(n)} = o(2^{g(n)})$?

3. Заполните табличку и поясните (особенно 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	$2^{n/2}$					
$n^{\log m}$	$m^{\log n}$					
$\log(n!)$	$\log(n^n)$					

   Здесь все буквы, кроме n, — константы.

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

   - Для каждого i найдите максимальное такое r_i, что $\sum_{j=i}^{r_i} a_j \le S$ за $\mathcal{O}(n)$ для каждого i.
   - Найдите за $\mathcal{O}(n)$ ответ задачи, если известны $r_1, \ldots, r_n$.
   - Докажите, что $r_i \le r_{i+1}$
   - Пользуясь предыдущим пунктом найдите все r_i за $\mathcal{O}(n)$.

5. Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{N}$.

   (a) За $\mathcal{O}(n)$ для каждого a_i найти самый правый из элементов, которые левее и меньше его.

   (b) За $\mathcal{O}(n)$ для каждого a_i найти самый левый из элементов, которые правее и меньше его.

   (c) За $\mathcal{O}(n)$ 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.

   (d) За $\mathcal{O}(n)$ 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.

6. 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.

7. *(только группа Антона)* Определить асимптотику $T(n) = 2 \cdot T(\lfloor \log n \rfloor) + 2^{\log^* n}$, где $\log^* n$ — итерированный логарифм.

посчитать сколько элемент входит в
запросы в отдельном листе

1.3 Дополнительные задачи

1. Упорядочите функции по скорости роста и обозначьте неравенства между соседями. Укажите, в каких неравенствах $f = o(g)$, а в каких $f = \Theta(g)$

$\log(\log^* n)$	$2^{log^* n}$	$(\sqrt{n})^{\log n}$	n^2	$n!$	$(\log n)!$
$(3/2)^n$	n^3	$\log^2 n$	$\log n!$	2^{2^n}	$n^{1/\log n}$
$\ln \ln n$	$\log^* n$	$n \cdot 2^n$	$n^{\log \log n}$	$\ln n$	1
$2^{\ln n}$	$(\log n)^{\log n}$	e^n	$4^{\log n}$	$(n+1)!$	$\sqrt{\log n}$
$\log^* \log n$	$2^{\sqrt{2 \log n}}$	n	2^n	$n \log n$	$2^{2^{n+1}}$

Примечание: $\log^*(n) = \begin{cases} 0 & \texttt{если } n \le 1; \\ 1 + \log^*(\log n) & \texttt{иначе.} \end{cases}$

2. Определить асимптотику (считайте, что при $n \le 100$ будет выполняться $T(n) = 100$).
   (a) $T(n) = 2 \cdot T(\lfloor \frac{n}{2} \rfloor + 17) + n$.
   (b) $T(n) = T(\alpha \cdot n) + T((1 - \alpha) \cdot n) + n$ для произвольной константы $\alpha \in (0, 1)$.
   (c) $T(n) = 4 \cdot T(\lfloor \frac{n}{2} \rfloor) + n^k$ для $k \in \{1, 2, 3\}$.
3. Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{Z}$. Найти l, r $(1 \le l \le r \le n)$ такие, что сумма $\sum_{i=l}^{r} a_i$ была бы максимальной. Задачу требуется решить за линейное от n время.
4. Дано число, представленное n цифрами в d-ичной записи без ведущих нулей. Из числа требуется вычеркнуть ровно k цифр так, чтобы результат был максимальным. Задачу требуется решить за линейное от n время.
5. Вам дан массив из n элементов и число k. Все числа лежат в отрезке $[1..n]$. Найдите такие l и r, что на отрезке $[l, r]$ встречается хотя бы k различных элементов, или сообщите, что такого отрезка нет. Если таких отрезков несколько, выберите тот из них, длина которого минимальна. Время работы $\mathcal{O}(n)$.
6. 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.
7. Дана квадратная матрица из нулей и единиц. Найти наибольший по площади подпрямоугольник, состоящий только из нулей за $\mathcal{O}(n^2)$.
8. Вам каждый день на протяжении некоторого времени поступает запрос «вырастет ли курс Apple на бирже», и у вас есть n советников, с которыми вы можете консультироваться. Вы отвечаете да или нет, и в конце каждого дня вам говорят, правильно ли вы ответили. Придумайте алгоритм, который сделает не более $10(\log n + m)$ ошибок, где m — число ошибок, которое сделает лучший советник (подсказка: назначьте советникам веса и изменяйте их в зависимости от правильности их ответов).
9. Придумайте расширяющийся массив с реальным (не амортизированным) временем добавления $\mathcal{O}(1)$.
10. Дан массив целых чисел от 1 до n длины $n + 1$, который нельзя модифицировать. Используя $\mathcal{O}(\log n)$ битов дополнительной памяти, найдите в массиве пару одинаковых чисел за $\mathcal{O}(n)$.
11. Дана последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n] = \{1, 2, \cdots, n\}$. Обозначим частоту появления элемента x через $f_\sigma[x] = |\{i | a_i = x\}|$. Известно, что $\exists_x f_\sigma[x] = 1$ и для всех остальных значений $y \ne x, f_\sigma[y] \equiv 0 \mod 2$. Требуется найти x за один проход по последовательности, используя $\mathcal{O}(\log n + \log m)$ бит памяти.
12. Дана последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Известно, что $\exists_x f_\sigma[x] > \frac{m}{2}$. Требуется найти x за один проход по последовательности, используя $\mathcal{O}(\log n + \log m)$ бит памяти.