Практика по алгоритмам

Александр Мишунин, Михаил Слабодкин*

Осень, 2018

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

1 Практика 1. Асимптотика

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.

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. **Асимметрия**
   (a) Правда ли, что $\min(f(n), g(n)) = \Theta(f(n) + g(n))$?
   (b) Правда ли, что $\max(f(n), g(n)) = \Theta(f(n) + g(n))$?
4. **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: $\forall f, g, h : (f \sim g) \wedge (g \sim h) \Rightarrow f \sim h$.
5. **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) Правда ли, что $\preceq$ — отношение предпорядка (рефлексивное и транзитивное)?
   (b) Правда ли, что $\preceq$ — отношение частичного порядка (+ антисимметричность)?
   (c) Правда ли, что $\preceq$ — отношение частичного порядка на классах эквивалентности по $\sim$?
6. Правда ли, что если $y(n)$ — монотонная неограниченная функция, и $f(n) = \mathcal{O}(g(n))$, то $f(\,y(n)\,) = \mathcal{O}(g(\,y(n)\,))$?
7. Требуется реализовать очередь с амортизированным временем работы всех операций $\mathcal{O}(1)$, используя $\mathcal{O}(1)$ стеков.

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

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

10. (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) $\sum_{k=1}^{n} \frac{1}{k} = \Omega(\log n)$?

11. Оцените время работы следующих программ:

    (a)
```
for (a = 1; a < n; a++)
  for (b = 0; b < n; b += 1)
    {}
```

    (b)
```
for (a = 1; a < n; a++)
  for (b = 0; b < n; b += a)
    {}
```

    (c) Найти такие $a, b, c \in \mathbb{N} : abc = n, a + b + c = \min$. Решение:
```
for (a = 1; a <= n; ++a)
  for (b = 1; a * b <= n; ++b)
    c = N / a / b, ... ;
```

    (d) Еще одно решение (c):
```
for (a = 1; a * a * a <= n; ++a)
  for (b = 1; b * b <= n; ++b)
    c = N / a / b, ... ;
```

    (e) И еще одно решение (c):
```
for (a = 1; a * a * a <= n; ++a)
  for (b = a; a * b * b <= n; ++b)
    c = N / a / b, ... ;
```

    (f) Дополнительный вопрос: что делает этот код?
```
a = 1, b = n;
while (a < b) {
  while (x[a] < M && a <= b) a++;
  while (x[b] > M && a <= b) b--;
  if (a <= b) swap(x[a++], x[b--]);
}
```

    (g) Дополнительный вопрос: а если бы вместо 2 было бы 1?

```
while (a >= 2)
  a = sqrt(a);
```

(h) Решето Эратосфена (пользуемся, что: $p_n \approx n \ln n$)

```
for (p = 2; p < n; p++)
  if (min_divisor[p] == 0) // is prime
    for (x = p + p; x < n; x += p)
      if (min_divisor[x] == 0)
        min_divisor[x] = p;
```

Дополнительные задачи

12. 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 $\mathcal{O}(n)$.

13. 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 $y \neq x, f_\sigma[y] \equiv 0 \mod 2$. Требуется найти x за один проход по последовательности, используя $\mathcal{O}(\log n + \log m)$ бит памяти.

14. Дана последовательность $\sigma =< a_1, a_2, \cdots, a_{n-1} >$, где каждый $a_i \in [n]$. Известно, что $\exists_x f_\sigma[x] = 0$ и для всех остальных значений $y \neq x, f_\sigma[y] = 1$. Требуется найти x за один проход по последовательности, используя $\mathcal{O}(\log n)$ бит памяти.

15. 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.

16. Дана последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Требуется проверить, правда ли, что $\exists_x f_\sigma[x] > \frac{m}{2}$, и если такой x есть, то найти его за один проход по последовательности. Докажите, что любое решение потребует $\Omega(m \cdot (\log n - \log m + 1))$ бит памяти.

17. Разрешим сделать два прохода по последовательности. Решите прошлую задачу за $\mathcal{O}(\log n + \log m)$ бит памяти.

18. Даны число k и последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Назовем элемент x k-частым, если $f_\sigma[x] > \frac{m}{k}$. Придумайте, как найти все k-частые элементы за два прохода и $\mathcal{O}(k \cdot (\log n + \log m))$ бит памяти.

19. Даны число k последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Требуется отсортировать данную последовательность за k проходов и $\mathcal{O}(\frac{m}{k} \log n)$ бит памяти. Докажите, что ее нельзя отсортировать ее за $o(\frac{m}{k} \log n)$ бит памяти и k проходов.

1.2 Домашнее задание

1. Эквивалентны ли следующие факты:

   - $f = \Theta(g)$
   - $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \mathcal{O}(1)$

2. 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.

3. Продолжим отношение “$\preceq$” на функциях до отношения на классах эквивалентности по отношению эквивалентности “$\sim$”, введённому на паре. Правда ли, что получится отношение *линейного порядка*? То есть $\forall x, y : (x \preceq y) \vee (y \preceq x)$.

4. Покажите, что: $g(n) = o(f(n)) \Rightarrow f(n) + g(n) = \Theta(f(n))$

5. Считайте здесь, что $\forall n : f(n) > 1 \wedge g(n) > 1$.

   (a) Правда ли, что $f(n) = o(g(n)) \Rightarrow \log f(n) = o(\log g(n))$?

   (b) Правда ли, что $f(n) = o(g(n)) \Rightarrow 2^{f(n)} = o(2^{g(n)})$?

6. 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.

1.3 Дополнительные задачи

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

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

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

8. 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 $\mathcal{O}(n)$.

9. 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 $y \neq x, f_\sigma[y] \equiv 0 \mod 2$. Требуется найти x за один проход по последовательности, используя $\mathcal{O}(\log n + \log m)$ бит памяти.

10. Дана последовательность $\sigma =< a_1, a_2, \cdots, a_{n-1} >$, где каждый $a_i \in [n]$. Известно, что $\exists_x f_\sigma[x] = 0$ и для всех остальных значений $y \neq x, f_\sigma[y] = 1$. Требуется найти x за один проход по последовательности, используя $\mathcal{O}(\log n)$ бит памяти.

11. 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.

12. Дана последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Требуется проверить, правда ли, что $\exists_x f_\sigma[x] > \frac{m}{2}$, и если такой x есть, то найти его за один проход по последовательности. Докажите, что любое решение потребует $\Omega(m \cdot (\log n - \log m + 1))$ бит памяти.

13. Разрешим сделать два прохода по последовательности. Решите прошлую задачу за $\mathcal{O}(\log n + \log m)$ бит памяти.

14. Даны число k и последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Назовем элемент x k-частым, если $f_\sigma[x] > \frac{m}{k}$. Придумайте, как найти все k-частые элементы за два прохода и $\mathcal{O}(k \cdot (\log n + \log m))$ бит памяти.

15. Даны число k последовательность $\sigma =< a_1, a_2, \cdots, a_m >$, где каждый $a_i \in [n]$. Требуется отсортировать данную последовательность за k проходов и $\mathcal{O}(\frac{m}{k} \log n)$ бит памяти. Докажите, что ее нельзя отсортировать ее за $o(\frac{m}{k} \log n)$ бит памяти и k проходов.