Практика по алгоритмам

Владислав Кораблинов, Алексей Лапенок, Александр Мишунин *

Осень, 2020

Содержание

1 Асимптотика **3**
1.1 Практика . . . 3
1.2 Домашнее задание . . . 6
1.3 Дополнительные задачи . . . 6

2 Линейные алгоритмы **8**
2.1 Практика . . . 8
2.2 Дополнительные задачи . . . 8
2.3 Домашнее задание . . . 10

3 Разделяй и властвуй **12**
3.1 Практика . . . 12
3.2 Домашнее задание . . . 14
3.3 Дополнительные задачи . . . 14

4 Сортировки и кучи **15**
4.1 Практика . . . 15
4.2 Домашнее задание . . . 17

5 Шустрая сортировка и порядковые статистики **19**
5.1 Практика . . . 19
5.2 Домашнее задание . . . 20
5.3 Дополнительные задачи . . . 21

6 Линейные сортировки и порядковые статистики **22**
6.1 Практика . . . 22
6.2 Домашнее задание . . . 23

7 Демоническое программирование **25**
7.1 Практика . . . 25
7.2 Домашнее задание . . . 27

8 Архидемоническое программирование **28**
8.1 Практика . . . 28
8.2 Домашнее задание . . . 29

9 Жадные алгоритмы **30**
9.1 Практика . . . 30
9.2 Домашнее задание . . . 31

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

10 Остовные деревья **33**
10.1 Практика . . . 33
10.2 Домашнее задание . . . 34

11 Система непересекающихся множеств **35**
11.1 Практика . . . 35
11.2 Домашнее задание . . . 36

12 DFS **37**
12.1 Практика . . . 37
12.2 Домашнее задание . . . 38

13 BFS **39**
13.1 Практика . . . 39
13.2 Домашнее задание . . . 40

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. **Несколько аргументов**
   Придумайте определение для $f(n, m) = \mathcal{O}(g(n, m))$.
4. **Асимметрия**
   (a) Правда ли, что $\min(f(n), g(n)) = \Theta(f(n) + g(n))$?
   (b) Правда ли, что $\max(f(n), g(n)) = \Theta(f(n) + g(n))$?
5. **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: $\forall f, g, h : (f \sim g) \wedge (g \sim h) \Rightarrow f \sim h$.
6. **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 $\sim$?
7. Правда ли, что если $y(n)$ — монотонная неограниченная функция, и $f(n) = \mathcal{O}(g(n))$, то $f(\,y(n)\,) = \mathcal{O}(g(\,y(n)\,))$?
8. (a) Требуется реализовать очередь с амортизированным временем работы всех операций $\mathcal{O}(1)$, используя $\mathcal{O}(1)$ стеков.

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

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

9. Считайте здесь, что функции здесь $\mathbb{N} \to \mathbb{N}$ и что $\forall n : f(n) > 1 \wedge g(n) > 1$.

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

10. 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

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

12. Дана последовательность $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, где каждый $a_i \in [n] = \{1, 2, \cdots, n\}$. Обозначим частоту появления элемента x через $f_\sigma[x] = |\{i \mid a_i = x\}|$. 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.

13. Дана последовательность $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, 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.

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

1.2 Домашнее задание

1. Эквивалентны ли следующие факты:
   - $f = \Theta(g)$
   - $\exists C, 0 < C < \infty : \lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = C$

2. (*кроме группы Лапенка*) 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. (*только группа Мишунина*) 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. (*кроме группы Мишунина*)
   (a) Правда ли, что $\min(f(n), g(n)) = \Theta(f(n) + g(n))$?
   (b) Правда ли, что $\max(f(n), g(n)) = \Theta(f(n) + g(n))$?

5. (*кроме группы Кораблинова*) Покажите, что: $g(n) = o(f(n)) \Rightarrow f(n) + g(n) = \Theta(f(n))$

6. (*кроме группы Мишунина*) Считайте здесь, что $\forall n : f(n) > 1 \wedge g(n) > 1$.
   (a) Правда ли, что $f(n) = \mathcal{O}(g(n)) \Rightarrow 2^{f(n)} = \mathcal{O}(2^{g(n)})$?
   (b) Правда ли, что $f(n) = o(g(n)) \Rightarrow \log f(n) = o(\log g(n))$?
   (c) Правда ли, что $f(n) = o(g(n)) \Rightarrow 2^{f(n)} = o(2^{g(n)})$?

7. 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, — положительные константы.

1.3 Дополнительные задачи

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

9. Дана последовательность $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, где каждый $a_i \in [n] = \{1, 2, \cdots, n\}$. Обозначим частоту появления элемента x через $f_\sigma[x] = |\{i \mid a_i = x\}|$. 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 $\sigma = \langle a_1, a_2, \cdots, a_{n-1} \rangle$, 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 $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, 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 $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, 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 $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, 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 $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, 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.

2 Линейные алгоритмы

2.1 Практика

1. 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.

2. 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.

3. Дан массив целых чисел a_i. Придумайте структуру данных, которая бы умела отвечать на online 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.

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.

5. 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.

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

7. 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.

8. Дана последовательность $a_1, a_2, \cdots, a_n \in \mathbb{N}$. Найти l, r $(1 \le l \le r \le n)$ 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.

9. (a) Найти в данном массиве число отрезков, содержащих ровно k единиц. $\mathcal{O}(n)$.

   (b) Найти отрезок максимальной длины без повторяющихся элементов. $\mathcal{O}(n)$, числа из $[1..n]$.

   (c) Найти в массиве `a` позиции `i`, `j`, `k`: `a[i] + a[j] + a[k] = S` за $\mathcal{O}(n^2)$.

2.2 Дополнительные задачи

10. Дана квадратная матрица из нулей и единиц. Найти наибольший по площади подпрямоугольник, состоящий только из нулей за $\mathcal{O}(n^2)$.

11. 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.

12. Дана строка s из скобок, а также число k. Для всех i проверить, что подстрока $s[i..i+k]$ является правильной скобочной последовательностью. $\mathcal{O}(n)$. Скобки только одного типа.

13. Вам поступает куча запросов online вычислить неизвестную функцию $F : X \to \{0, 1\}$, после каждого запроса вам говорят правильно ли вы ответили. У вас есть набор детерминированных функций-оракулов $f_1 \ldots f_n$, решающих задачу с ошибками, про которые вы больше ничего не знаете. Придумайте алгоритм, который делает $\mathcal{O}(\log n + m)$ ошибок, где m — число ошибок, которое делает лучший оракул (подсказка: назначьте оракулам веса).

14. Вам дана последовательность из n чисел. Нужно offline ответить на m запросов вида «количество различных чисел на отрезке l, r» за $\mathcal{O}((m+n)\sqrt{n})$.

2.3 Домашнее задание

1. 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.

2. Дано число, представленное n цифрами в d-ичной записи без ведущих нулей. Из числа требуется вычеркнуть ровно k цифр так, чтобы результат был максимальным. $\mathcal{O}(n)$, d и k — не константы.

3. *(только группа Кораблинова)* 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. *(только группа Мишунина)* Дан массив целых чисел длины n. Найдите подотрезок длины от L до R с максимальной суммой. $\mathcal{O}(n)$.

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

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

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

   (c) *(только группы Кораблинова и Лапенка)* 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) *(только группы Кораблинова и Лапенка)* 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. *(кроме группы Мишунина)* Дан массив длины n с целыми числами a_i, $1 \le a_i \le n$ и число k. 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 $\mathcal{O}(n)$.

7. *(только группа Мишунина)* Дан массив длины n с целыми числами a_i, $-n \le a_i \le n$, и число k. Найдите подотрезок с максимальной суммой, на котором встречается хотя бы k различных чисел. $\mathcal{O}(n)$.

8. *(кроме группы Кораблинова)* Дан массив длины n с целыми числами a_i, $-n \le a_i \le n$. Найдите подотрезок с максимальной суммой, длины от L до R, содержащий от A до B различных чисел, за $\mathcal{O}(n)$.

9. *(кроме группы Кораблинова)* Вам дана строка из трёх типов скобок. Найдите самую длинную ее подстроку, являющуюся правильной скобочной последовательностью. $\mathcal{O}(n)$.

10. Дано двоичное число, заданное в виде битового массива a длины n (от младших битов к старшим), к которому затем m раз прибавляют единицу простейшим алгоритмом:

```
int carry = 1;
for (int i = 0; carry > 0 && i < n; i++)
{
  carry += a[i];
  a[i] = carry % 2;
  carry = carry / 2;
}
```

    Числа выбраны так, что переполнения не происходит. Докажите методом потенциалов, что суммарное время работы будет $\mathcal{O}(n+m)$.

Дополнительные задачи

11. За один проход со стеком найти ближайшие “$<$” слева и “$\leq$” справа.
12. 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.
13. Дана квадратная матрица из нулей и единиц. Найти наибольший по площади подпрямоугольник, состоящий только из нулей за $\mathcal{O}(n^2)$.

3 Разделяй и властвуй

3.1 Практика

Пусть $a \geq 1$ и $b > 1$ — константы, $f(n)$ — функция, $T(n)$ определено при неотрицательных n формулой

$$T(n) = aT(n/b) + f(n),$$

где под n/b понимается либо $\lceil n/b \rceil$, либо $\lfloor n/b \rfloor$. Тогда

- если $f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$ для некоторого $\epsilon > 0$, то $T(n) = \Theta(n^{\log_b a})$;
- если $f(n) = \Theta(n^{\log_b a})$, то $T(n) = \Theta(n^{\log_b a} \log n)$;
- если $f(n) = \Omega(n^{\log_b a + \epsilon})$ для некоторого $\epsilon > 0$ и если $af(n/b) \leq cf(n)$ для некоторой константы $c < 1$ и достаточно больших n, то $T(n) = \Theta(f(n))$.

1. Дана монотонно неубывающая функция $[1 \dots n] \to \{0, 1\}$. Напишите псевдокод, находящий последний 0 и первую 1 за $\mathcal{O}(\log n)$ вызовов функции.
2. Сделайте предподсчет за $\mathcal{O}(n \log n)$, чтобы за $\mathcal{O}(\log n)$ online отвечать на запрос “сколько раз число x встречается на отрезке $[l..r]$”?
3. Есть n веревок, каждая имеет целую длинну l_i, которые можно резать. Нужно получить k одинаковых кусков максимальной целочисленной длины (также могут остаться неиспользованные обрезки). $\mathcal{O}(n \log l_{\max})$.
4. Определить асимптотику.
   (a) $T(n) = T(\lceil \frac{n}{2} \rceil) + 1$
   (b) $T(n) = T(a) + T(n - a) + n$ для произвольной константы a.
   (c) $T(n) = 2 \cdot T(\lfloor \frac{n}{2} \rfloor + 17) + n$.
   (d) $T(n) = T(\alpha \cdot n) + T((1 - \alpha) \cdot n) + n$ для произвольной константы $\alpha \in (0, 1)$.
   (e) $T(n) = 4 \cdot T(\lfloor \frac{n}{2} \rfloor) + n^k$ для $k \in \{1, 2, 3\}$.
   (f) $T(n) = 2 \cdot T(\sqrt{n}) + \log n$
5. Есть k отсортированных массивов. В сумме массивы содержат n элементов. Слить массивы за $\mathcal{O}(n \log k)$.
6. Инверсией в массиве чисел $a[\dots]$ называется такая пара индексов i, j, что $i < j$, но $a_i > a_j$. Дан массив из n различных элементов. Требуется найти число инверсий за $\mathcal{O}(n \log n)$.
7. Даны два сортированных массива длины n, которые нельзя модифицировать. Найдите k-ю порядковую статистику в объединении массивов, используя $\mathcal{O}(1)$ дополнительной памяти.
   (a) За $\mathcal{O}(\log^2 n)$.
   (b) За $\mathcal{O}(\log n)$.
8. Даны два массива a и b длины n, сгенерировать все попарные суммы $a_i + b_j$ в сортированном порядке.
   (a) За $\mathcal{O}(n^2 \log n)$.
   (b) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.
   (c) За $\mathcal{O}(n^2 \log n)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.
   (d) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(1)$ дополнительной памяти.
9. Дан массив из $2n$ чисел. Найти минимальное и максимальное за $3n - 2$ сравнения.
10. Найти второй максимум в массиве за $n + \mathcal{O}(\log n)$ сравнений.
11. Найти отрезок с максимальным средним арифметическим, при этом длины от L до R. $\mathcal{O}(n \log n)$.

Дополнительные задачи

12. Найти количество `AVL` деревьев высоты h из n вершин по простому модулю p за $\mathcal{O}(hn^{\log_2 3})$ (`AVL` дерево — корневое двоичное дерево, в котором у каждой вершины высоты двух дочерних поддеревьев отличаются не более, чем на 1).

13. Дано множество из n точек на плоскости. Найти пару ближайших точек за $\mathcal{O}(n \log n)$.

14. Даны два массива из положительных чисел a и b. $|a| = |b| = n \le 10^5$.
    Выбрать массив p: k различных чисел от 1 до n так, чтобы $\frac{\sum_{i=1}^{k} a_{p_i}}{\sum_{i=1}^{k} b_{p_i}} \to \max$.

3.2 Домашнее задание

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

2. *(только группа Кораблинова)* Определить асимптотику.

   (a) $T(n) = T(\lceil \frac{n}{2} \rceil) + 1$

   (b) $T(n) = T(a) + T(n - a) + n$ для произвольной константы a.

   (c) $T(n) = T(\alpha \cdot n) + T((1 - \alpha) \cdot n) + n$ для произвольной константы $\alpha \in (0, 1)$.

   (d) $T(n) = 2 \cdot T(\lfloor \frac{n}{2} \rfloor + 17) + n$.

3. Есть m стойл с координатами $x_1, \ldots, x_m$ и n коров. Расставить коров по стойлам (не более одной в стойло) так, чтобы минимальное расстояние между коровами было максимально. $\mathcal{O}(m(\log m + \log x_{\max}))$.

4. *(кроме группы Лапенка)* Докажите, что для поиска максимума в массиве различных чисел потребуется как минимум $n - 1$ сравнение.

5. Найти второй максимум в массиве за $n + \mathcal{O}(\log n)$ сравнений.

6. Даны два сортированных массива длины n, которые нельзя модифицировать. Найдите k-ю порядковую статистику в объединении массивов (то есть элемент, находившийся бы на k-ой позиции если бы массивы слили), используя $\mathcal{O}(1)$ дополнительной памяти.

   (a) За $\mathcal{O}(\log^2 n)$.

   (b) За $\mathcal{O}(\log n)$.

7. *(только группа Мишунина)* Структура данных «файл последовательного доступа» поддерживает следующие операции за $\mathcal{O}(1)$:

   - *Read()*: чтение числа из файла на текущей позиции и перевод позиции вперёд на 1 элемент.
   - *Write(x)*: запись числа в файл в текущую позицию и перевод позиции вперёд на 1 элемент.
   - *Rewind()*: перевод позиции на начало файла.

   Требуется отсортировать файл за $\mathcal{O}(n \log n)$ используя $\mathcal{O}(1)$ дополнительных ячеек памяти и $\mathcal{O}(1)$ дополнительных файлов.

8. *(кроме группы Кораблинова)* Дано $2n - 1$ коробок с чёрными и белыми шарами. В i-ой коробке находится w_i белых и b_i чёрных шаров. Всего в коробках находится W белых и B чёрных шаров. Требуется выбрать n коробок таким образом, чтобы суммарное число белых шаров в них было не менее $\frac{W}{2}$, а чёрных не менее $\frac{B}{2}$. Решить за $\mathcal{O}(n \log n)$.

3.3 Дополнительные задачи

1. Дан массив из $2n$ различных чисел. Найдите минимальное и максимальное за $3n - 2$ сравнения и докажите, что это точная нижняя оценка, то есть меньшего количества сравнений может не хватить.

2. Найдите второй максимум в массиве за $n + \lceil \log_2 n \rceil - 2$ сравнение и докажите, что это точная нижняя оценка, то есть меньшего количества сравнений может не хватить.

4 Сортировки и кучи

4.1 Практика

1. Покажите, что любая сортировка, которая верно работает хотя бы на доле $\frac{1}{100^n}$ от всех перестановок, не может работать за $o(n \log n)$ на всех тестах.

2. Модифицируйте операцию `SiftUp` для бинарной кучи так, чтобы она по-прежнему работала за $\mathcal{O}(\log n)$, но при этом делала лишь $\mathcal{O}(\log \log n)$ сравнений.

3. Модифицируйте операцию `SiftDown` для бинарной кучи так, чтобы она по-прежнему работала за $\mathcal{O}(\log n)$, но при этом делала лишь $\log_2 n + \mathcal{O}(\log \log n)$ сравнений.

4. Придумайте детерминированную структуру данных на основе бинарной кучи, которая умеет делать `Insert(x)`, `DeleteMedian()`, за $\mathcal{O}(\log n)$, `GetMedian()` за $O(1)$.

5. Дана обычная бинарная куча (с минимумом в голове), требуется узнать k-й минимум.

   (a) $\mathcal{O}(k \log n)$

   (b) $\mathcal{O}(k^2)$

   (c) $\mathcal{O}(k \log k)$

6. Даны два массива a и b длины n, сгенерировать все попарные суммы $a_i + b_j$ в сортированном порядке.

   (a) За $\mathcal{O}(n^2 \log n)$.

   (b) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.

   (c) За $\mathcal{O}(n^2 \log n)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.

   (d) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(1)$ дополнительной памяти. Входные массивы отсортированы.

7. Дан массив длины n, в котором встречаются $m \le n$ различных элементов.

   (a) Пусть зафиксирован набор частот элементов $p_i > 0, i = 1 \dots m$. Докажите нижнюю оценку $n\left(\sum_{i=1}^{m} p_i \log \frac{1}{p_i}\right) - n \log e$ на число сравнений в худшем случае при сортировке сравнениями. Полезный факт: $n \ln n \ge \ln n! = n \ln n - n + \mathcal{O}(\ln n)$.

   (b) Докажите нижнюю оценку $n \log \frac{m}{e}$ на число сравнений в худшем случае при сортировке сравнениями в пределе $n \gg m$.

8. Дано бинарное дерево: $Tree ::= Node(Tree, Tree)|Empty$ (эта запись означает, что дерево — это либо вершина с парой потомков-деревьев, либо особое значение $Empty$). Определим функцию $\mathbf{rank}(x)$ следующим образом:

   - $\mathbf{rank}(Empty) = 0$
   - $\mathbf{rank}(Node(left, right)) = min(\mathbf{rank}(left), \mathbf{rank}(right)) + 1$.

   Назовём бинарное дерево *скошенным влево (левацким)*, если для его вершин выполнено следующее свойство:

   $$\forall_{x=Node(left,right)} \mathbf{rank}(left) \ge \mathbf{rank}(right).$$

   Скошенная влево (левацкая) куча — это скошенное влево дерево, в вершинах которого хранятся данные, для которых выполнено свойство кучи.

   (a) Докажите, что для любого скошенного влево дерева $|T| \ge 2^{\mathbf{rank}(T)} - 1$ ($|T|$ обозначает количество вершин в дереве T).

   (b) Придумайте, как слить две скошенные влево кучи H_1 и H_2 за время $\mathcal{O}(\log |H_1| + \log |H_2|)$.

   (c) Придумайте, как используя операцию слияния, построенную на предыдущем шаге, реализовать операции:

- `Insert(x)` — добавление элемента x в кучу,
- `ExtractMin()` — удаление минимального элемента из кучи.

9. Пусть B_n (биномиальное дерево порядка n) определено следующим образом:

   - при $n = 0$ это дерево из одной вершины.
   - при $n > 0$ это B_{n-1}, которому первым ребенком подвешено еще одно B_{n-1}.

   (a) Докажите, что B_n имеет высоту n.

   (b) Докажите, что в B_n содержится ровно 2^n вершин.

   (c) Определим биномиальную кучу как набор биномиальных деревьев, в котором нет двух деревьев одного порядка. Покажите, что для любого n существует биномиальная куча с n вершинами.

   (d) Пусть на всех деревьях биномиальной кучи выполняется свойство кучи (min в голове). Придумайте `GetMin` за $\mathcal{O}(\log n)$.

   (e) Придумайте `Merge` за $\mathcal{O}(\log n)$.

   (f) Придумайте `Add` за $\mathcal{O}(\log n)$.

   (g) Придумайте `ExtractMin` за $\mathcal{O}(\log n)$.

   (h) Придумайте `DecreaseKey` по ссылке на узел. $\mathcal{O}(\log n)$.

   (i) Придумайте `Delete` по ссылке на узел. $\mathcal{O}(\log n)$.

10. Покажите, что n операций `Add` подряд в пустую биномиальную кучу работают за $\mathcal{O}(n)$.

Дополнительные задачи

11. Предъявите последовательность действий, при которой `LeftistHeap` из пустой станет бамбуком (цепочкой из вершин, у каждой один ребенок).

12. Куча хранится в массиве длины n. Родитель p хранит детей в ячейках $2 \cdot p + 1$ и $2 \cdot p + 2$. Алгоритм приступает к сортировке. Сортировка устроена следующим образом.

    - Поменять первый и последний элемент кучи местами.
    - Уменьшить размер кучи на единицу.
    - Запустить `SiftDown` на первом элементе.

    `SiftDown` меняет родителя с наибольшим ребенком (при условии, что ребенок больше родителя)и запускается рекурсивно. Требуется придумать алгоритм, который по n выдаёт перестановку чисел от 1 до n, которая является корректной кучей и приводит к максимальному количеству вызовов `SiftDown` при сортировке. Время работы — $\mathcal{O}(n \log n)$.

4.2 Домашнее задание

1. *(только группа Кораблинова)* Дана обычная бинарная куча (с минимумом в голове), требуется узнать k-й минимум за $\mathcal{O}(k \log k)$.

2. *(только группа Кораблинова)* Есть k отсортированных массивов. В сумме массивы содержат n элементов. Слить массивы в один отсортированный за $\mathcal{O}(n \log k)$.

3. *(только группа Кораблинова)* Даны два массива a и b длины n, сгенерировать все попарные суммы $a_i + b_j$ в сортированном порядке.

   (a) За $\mathcal{O}(n^2 \log n)$.

   (b) За $\mathcal{O}(n^2 \log n)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.

   (c) *(доп.)* За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(1)$ дополнительной памяти. Входные массивы отсортированы.

4. В свободное время Анка-пулемётчица любит сортировать патроны по серийным номерам. Вот и сейчас она только разложила патроны на столе в строго отсортированном порядке, как Иван Васильевич распахнул дверь с такой силой, что все патроны на столе подпрыгнули и немного перемешались. Оставив ценные указания, Иван Васильевич отправился восвояси. Как оказалось, патроны перемешались не сильно. Каждый патрон отклонился от своей позиции не более чем на k. Всего патронов n. Помогите Анке отсортировать патроны.

   (a) Отсортируйте патроны за $\mathcal{O}(nk)$.

   (b) Отсортируйте патроны за $\mathcal{O}(n + I)$, где I — число инверсий.

   (c) Докажите нижнюю оценку на время сортировки $\Omega(n \log k)$.

   (d) Отсортируйте патроны за $\mathcal{O}(n \log k)$.

5. Рассмотрим бинарную скошенную систему исчисления. На каждой позиции в скошенной записи числа может стоять цифра 0, 1 или 2. Число $\overline{a_k a_{k-1} \cdots a_2 a_1}$ в скошенной системе переводится в десятичную по формуле $\sum_{i=1}^{k} a_i \cdot (2^i - 1)$.

   В скошенной системе счисления есть два ограничения: цифра 2 может встречаться в записи не более одного раза; все цифры следующих меньших разрядов равны нулю. Пример первых чисел: 1, 2, 10, 11, 12, 20, 100, 101...

   (a) Докажите, что каждое неотрицательное целое число имеет единственное возможное представление в скошенной системе счисления.

   (b) Придумайте, как увеличить число в скошенной системе на единицу за $\mathcal{O}(1)$.

6. *(только группа Лапенка)* Как можно модифицировать биномиальную кучу, чтобы `Insert` выполнялось за истинное $O(1)$, а время работы остальных операций не поменялась?

7. *(кроме группы Кораблинова)* Определим структуру данных “скошенный список”. Список длины n строится так:

   - запишем число n в скошенной системе счисления: $\overline{a_k a_{k-1} \cdots a_2 a_1}$
   - для каждого i смотрим в соответствующую позицию скошенной записи числа n и создаём a_i полных двоичных деревьев высоты i
   - размещаем n элементов списка: сперва выбираем дерево в порядке возрастания высоты, а внутри конкретного дерева размещаем в порядке обхода в глубину: «корень, левый ребёнок, правый ребёнок»

   Примеры скошенных списков длин $1, 2, 3, 4, 5$:

Рис. 1: Лист [a] (число: 1)

Рис. 2: Лист [a b] (число: 2)

Рис. 3: Лист [a b c] (число: 10)

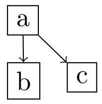


Рис. 4: Лист [a b c d] (число: 11)

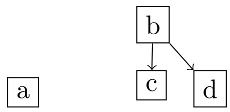
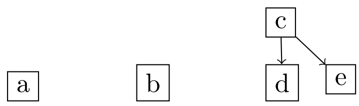


Рис. 5: Лист [a b c d e] (число: 12)



Придумайте, как реализовать следующие операции со списком длины n:

(a) Добавление элемента в начало списка за $\mathcal{O}(1)$.

(b) Доступ к i-му элементу за $\mathcal{O}(\log n)$.

(c) Получить скошенный список из k последних элементов данного скошенного списка за $\mathcal{O}(\log n)$.

Дополнительные задачи

8. Куча хранится в массиве длины n. Родитель p хранит детей в ячейках $2 \cdot p + 1$ и $2 \cdot p + 2$. Алгоритм приступает к сортировке. Сортировка устроена следующим образом.

   - Поменять первый и последний элемент кучи местами.
   - Уменьшить размер кучи на единицу.
   - Запустить `SiftDown` на первом элементе.

   `SiftDown` меняет родителя с наибольшим ребенком (при условии, что ребенок больше родителя)и запускается рекурсивно. Требуется придумать алгоритм, который по n выдаёт перестановку чисел от 1 до n, которая является корректной кучей и приводит к максимальному количеству вызовов `SiftDown` при сортировке. Время работы — $\mathcal{O}(n \log n)$.

5 Шустрая сортировка и порядковые статистики

5.1 Практика

1. (a) Приведите вероятностный алгоритм поиска медианы в массиве различных чисел со средним временем работы $\mathcal{O}(n)$

   (b) Приведите детерминированный алгоритм поиска медианы в массиве различных чисел с гарантированным временем работы $\mathcal{O}(n)$

2. Оцените время работы детерминированного алгоритма поиска порядковой статистики, если вместо пятерок разбивать элементы на

   (a) семерки.

   (b) тройки.

3. Придумайте, как добиться от `QuickSort` времени $\mathcal{O}(n \log n)$ в худшем случае.

4. Робот Иван Семеныч пробует пирожки. Содержимое пирожков делится на три типа. Всего пирожков n. Каждый пирожок можно попробовать не более одного раза. Любые два пирожка можно поменять местами. Память у робота маленькая, $\mathcal{O}(\log n)$ бит. Помогите Ивану Семенычу отсортировать пирожки по типу: сначала первый, потом второй, потом третий. Сортировка должна работать за линейное время.

5. Дан набор из n пар гаек и болтов, в разных парах размеры гаек и болтов различны. Гайки и болты перемешаны. Требуется для каждой гайки найти соответствующий болт. Сравнивать можно только болты с гайками (сравнить две гайки между собой, или два болта между собой — невозможно). $\mathcal{O}(n \log n)$.

6. Пусть задан массив A из $n = a \cdot k$ различных чисел. Требуется разбить массив на k частей по a элементов в каждой так, чтобы любой элемент части i был бы меньше любого элемента части $i+1$ ($\forall i \in [1, k-1]$). $\mathcal{O}(n \log k)$.

7. Дан массив из $2 \cdot n - 1$ числа, который нельзя модифицировать. Есть дополнительная память на $n+1$ элемент массива и ещё $\mathcal{O}(1)$ сверху. Требуется найти медиану за $\mathcal{O}(n \log n)$.

8. Дана последовательность из n чисел, нужно за один проход и $\mathcal{O}(n)$ времени найти в ней k минимумов, используя $\mathcal{O}(k)$ дополнительной памяти.

9. Даны массив из n чисел и m чисел $p_1, p_2, \ldots p_m$, нужно за $\mathcal{O}(n \log m + m)$ для каждого i найти p_i-ую порядковую статистику.

10. Даны два массива из положительных чисел a и b, $|a| = |b| = n$. Выбрать массив p из k различных чисел от 1 до n так, чтобы $\frac{\sum_{i=1}^{k} a_{p_i}}{\sum_{i=1}^{k} b_{p_i}} \to \max$. $\mathcal{O}(n \log n)$.

11. На прямой расположено n точек p_1, p_2, $\ldots$, p_n (p_i в произвольном порядке). Каждая точка имеет вес $w_i \geq 0$. Требуется найти точку q: $\sum_i [w_i \cdot |p_i - q|] \to \min$. $\mathcal{O}(n)$ в среднем.

5.2 Домашнее задание

1. Дан массив `A[1..n]` из n различных чисел. Массив не обязательно отсортирован. Требуется найти k ближайших к медиане элементов за линейное время. Решить для двух метрик.

   (a) По позиции в отсортированном массиве.

   $$d(x, \texttt{median}) = |\texttt{pos}(x) - \texttt{pos}(\texttt{median})|,$$

   где $\texttt{pos}(x)$ — позиция элемента x в отсортированном массиве.

   (b) По значению.

   $$d(x, \texttt{median}) = |x - \texttt{median}|.$$

2. В матрице Q из натуральных чисел размера $N \times N$ найти подматрицу размера $H \times W$ с максимальной медианой. H, W — нечётные.

   (a) $\mathcal{O}(N^2 \log Q_{\max})$. Здесь $Q_{\max}$ — максимальный элемент матрицы.

   (b) $\mathcal{O}(N^2 \log N)$.

3. *(только группа Кораблинова)* Дана последовательность из n чисел, нужно за один проход и $\mathcal{O}(n)$ времени найти в ней k минимумов, используя $\mathcal{O}(k)$ дополнительной памяти.

4. *(только группа Мишунина)* На прямой расположено n точек $p_1, p_2, \ldots, p_n$ (p_i в произвольном порядке). Каждая точка имеет вес $w_i \geq 0$. Требуется найти точку q: $\sum_i [w_i \cdot |p_i - q|] \to \min$. $\mathcal{O}(n)$ в среднем.

5. Будем называть массив `a[0..n-1]` *отсортированным на* 90%, или *почти отсортированным*, если из него можно так вычеркнуть 10% элементов, чтобы оставшиеся оказались в отсортированном порядке.

   Попытаемся построить эффективный алгоритм, по входному массиву отличающий два случая:

   - массив полностью отсортирован
   - массив **не** почти отсортирован

   Будем рассматривать только массивы, все элементы которых различны.

   Рассмотрим следующий алгоритм.

```
def binary_search(a, key, left, right):
    if left == right - 1:
        return left
    else:
        mid = left + (right - left) / 2
        if key < a[mid]:
            return binary_search(a, key, left, mid)
        else:
            return binary_search(a, key, mid, right)


def check-sort(a):
    for r in range(0, k):
        i = random.range(0, n)
        j = binary_search(a, a[i], 0, len( a ))
        if i != j:
            return False
    return True
```

   Докажите, что при выборе некоторого k данный алгоритм

- выдаёт всегда `True` для отсортированного массива;
- выдает `False` с вероятностью хотя бы $\frac{2}{3}$, если массив не отсортирован на 90%.

Оцените значения k, для которых это верно. (Засчитывание задачи зависит от точности оценки. Например, ответ $k > 0$ не будет засчитан)

5.3 Дополнительные задачи

6. Даны два массива из положительных чисел a и b, $|a| = |b| = n$. Выбрать массив p из k различных чисел от 1 до n так, чтобы $\frac{\sum_{i=1}^{k} a_{p_i}}{\sum_{i=1}^{k} b_{p_i}} \to \max$.

7. Пусть алгоритм A находит i-ый элемент, используя только попарные сравнения элементов. Покажите, что, используя результаты только этих сравнений, можно найти все элементы, меньшие i-ого, и все элементы, большие i-ого.

6 Линейные сортировки и порядковые статистики

6.1 Практика

1. Даны массив из n чисел и m чисел $p_1, p_2, \dots p_m$, нужно за $\mathcal{O}(n \log m + m)$ для каждого i найти p_i-ую порядковую статистику.

2. Дан массив из n чисел от 1 до k, разработайте структуру данных, которая за $\mathcal{O}(1)$ отвечает на запросы вида «Сколько в массиве элементов от a до b?». Время на предподсчет $O(n + k)$.

3. Как с помощью цифровой сортировки сортировать строки над константным алфавитом в лексикографическом порядке за $\mathcal{O}(\sum |s_i|)$?

4. Есть массив, состоящий из n натуральных чисел. Найти минимальное натуральное число, которого нет в массиве, за $\mathcal{O}(n)$.

5. Даны два массива a и b одинаковой длины. Найти такую перестановку p, что

   (a) $\sum\limits_{i=1}^{n} a_{p_i} b_i \to \max;$

   (b) $\sum\limits_{i=1}^{n} a_{p_i} b_i \to \min.$

6. На прямой живут n друзей, i-й друг живет в точке x_i. Они хотят встретиться в одной точке. Помогите им найти такую точку, чтобы:

   (a) сумма квадратов расстояний, которое они пройдут, была бы минимальна.

   (b) сумма расстояний, которое они пройдут, была бы минимальна.

7. Даны n точек на плоскости (x_i, y_i). Найти точку (x^*, y^*):

   (a) $\max\limits_i \Big[\max(|x_i - x^*|, |y_i - y^*|)\Big] \to \min$

   (b) $\max\limits_i \Big[|x_i - x^*| + |y_i - y^*|\Big] \to \min$

   (c) $\sum\limits_i \Big[(x_i - x^*)^2 + (y_i - y^*)^2\Big] \to \min$

   (d) $\sum\limits_i \Big[|x_i - x^*| + |y_i - y^*|\Big] \to \min$

   (e) $\sum\limits_i \Big[\max(|x_i - x^*|, |y_i - y^*|)\Big] \to \min$

8. На прямой расположено n точек p_1, p_2, $\dots$, p_n (p_i в произвольном порядке). Каждая точка имеет вес $w_i \geq 0$. Требуется найти точку $q\colon \sum_i [w_i \cdot |p_i - q|] \to \min$. $\mathcal{O}(n)$ в среднем.

9. На прямой расположено n точек p_1, p_2, $\dots$, p_n. Каждая точка имеет вес $w_i \geq 0$. Выбрать две точки $q_1, q_2\colon \sum\limits_i w_i \min(|p_i - q_1|, |p_i - q_2|) \to \min$.

10. Пусть мы хотим показать, что некоторая сортирующая сеть сливает два отсортированных массива в один. Покажите, что для этого достаточно показать, что она сливает массивы, состоящие из 0 и 1.

6.2 Домашнее задание

1. Есть улица длины l, которая освещается $n > 0$ фонарями, i-й фонарь находится в точке с вещественной координатой a_i. Фонарь освещает все точки улицы, которые находятся от него на расстоянии не больше d, где d — некоторое положительное число, общее для всех фонарей. Найдите минимальное d, при котором вся улица освещена. $\mathcal{O}(n)$, a_i не отсортированы.

2. *(кроме группы Кораблинова)* Пусть w — степень двойки, не превосходящая разрядности машины. Определим структуру данных `T<w>`, представляющую множество w-битных целых ключей, следующим образом: `T<1>` определим тривиально, а для больших w `T<w>` может быть либо пустой, либо содержать:

   - массив длины $2^{\frac{w}{2}}$ из `T<w/2>`
   - множество непустых индексов этого массива в `T<w/2>`
   - еще $\mathcal{O}(1)$ дополнительной информации.

   (a) Придумайте `insert` и `delete` для `T<w>` за $\mathcal{O}(\log w)$.

   (b) Придумайте `findGE(x)` — поиск минимального ключа $\geq x$ за $\mathcal{O}(\log w)$.

   (c) Придумайте, как отсортировать массив из n w-битных чисел за $\mathcal{O}(n \log w)$.

3. Даны n точек на плоскости (x_i, y_i) с весами $w_i \geq 0$. Найти точку (x^*, y^*):

   (a) $\sum\limits_i \Big[w_i(|x_i - x^*| + |y_i - y^*|)\Big] \to \min$

   (b) $\max\limits_i \Big[w_i(|x_i - x^*| + |y_i - y^*|)\Big] \to \min.$

4. Возьмем массив a из n элементов, каждый из которых — это число 1 до n. Циклический сдвиг номер i - это последовательность a_i, a_{i+1}, ..., a_{n1}, a_0, a_1, ..., a_{i1}

   (a) Предложите алгоритм сортировки циклических сдвигов в лексикографическом порядке за время $O(n^2)$.

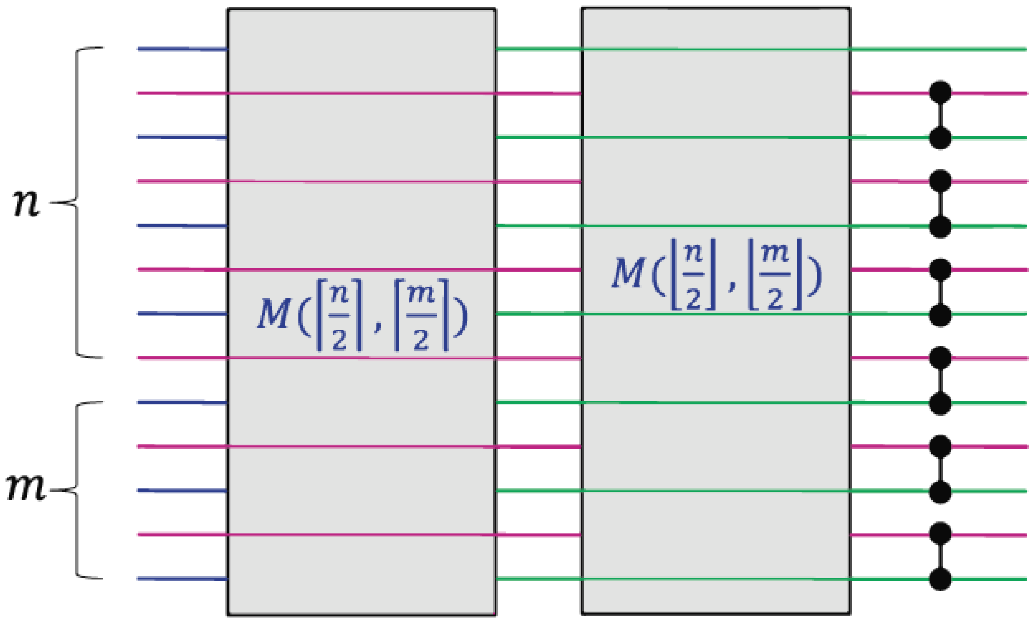
   (b) Предложите алгоритм сортировки циклических сдвигов за $O(n \log n)$. Указание: зная порядок на подстроках длины L порядок на подстроках длины $2L$ можно восстановить за $O(n)$.

5. *(только группа Кораблинова)* На прямой расположено n точек $p_1, p_2, \ldots, p_n$ (p_i в произвольном порядке). Каждая точка имеет вес $w_i \geq 0$. Требуется найти точку q: $\sum_i [w_i \cdot |p_i - q|] \to \min$. $\mathcal{O}(n)$ в среднем.

6. На прямой расположено n точек $p_1, p_2, \ldots, p_n$ (p_i отсортированы). Каждая точка имеет вес $w_i \geq 0$. Выбрать две точки q_1, q_2: $\sum\limits_i w_i \min(|p_i - q_1|, |p_i - q_2|) \to \min$. $\mathcal{O}(n)$.

Дополнительные задачи

7. Будем строить сортирующую сеть $M(n, m)$ для слияния двух отсортированных массивов размера n и m следующим образом. Разделим элементы обоих массивов на четные и нечетные, рекурсивно сольем отдельно четные, отдельно нечетные, в конце добавим компараторы для элементов $2k$ и $2k + 1$ (см картинку):



(a) Покажите, что такая сеть действительно сливает два отсортированных массива.

(b) Каково будет общее число компараторов в такой сети?

(c) Какова будет глубина такой сети?

7 Демоническое программирование

7.1 Практика

1. Найдите максимальную возрастающую подпоследовательность за $\mathcal{O}(n \log n)$. Подсказка: используйте дополнительный массив, в котором на i-той позиции стоит минимальный элемент, на который оканчивается некоторая возрастающая подпоследовательность длины i.

2. Даны две последовательности длины n. Придумайте, как найти наидлиннейшую общую подпоследовательность этих последовательностей.

   (a) За $\mathcal{O}(n^2)$.

   (b) За $\mathcal{O}(n \log n)$, в случае, если в одной из последовательностей все элементы различны.

3. Даны n предметов. У каждого заданы целый вес w_i и цена v_i. Найти max цену, которую можно набрать предметами суммарного веса $\leq S$. Время $\mathcal{O}(nS)$, память $\mathcal{O}(S)$.

   (a) Каждый предмет можно брать один раз.

   (b) Каждый предмет можно брать сколько угодно раз.

   (c) Каждый предмет можно брать сколько угодно раз. Восстановить ответ.

   (d) Каждый предмет можно брать один раз + набрать ровно S. Цен нет. Восстановить ответ.

   (e) Каждый предмет можно брать один раз. Восстановить ответ.

4. Дано рекуррентное соотношение:

   (a)

$$\begin{aligned} a_n &= a_{n-1} + 2b_{n-1} \\ b_n &= 10b_{n-1} - a_{n-1} \end{aligned}$$

   (b)

$$\begin{aligned} a_n &= a_{n-1} + 2c_{n-1} + 1 \\ b_n &= 5 - c_{n-1} \\ c_n &= c_{n-2} - b_{n-1} \end{aligned}$$

   Известны $a_0, a_1, b_0, b_1, c_0, c_1$. Найти a_n, b_n, c_n по модулю p за $\mathcal{O}(\log n)$.

5. Дан выпуклый n-угольник. Каждая диагональ n-угольника, соединяющая вершины i и j, имеет вес w_{ij}. Вес триангуляции многоугольника есть сумма весов диагоналей, которые в ней проведены. Найти триангуляцию с минимальным весом за $\mathcal{O}(n^3)$.

6. Судно атакуют пираты. Для каждого пирата известны его азимут a_i и время t_i, через которое пират приплывет и совершит непотребство. Однако, у судна есть лазерная пушка, которой оно защищается. У пушки есть начальный азимут a и угловая скорость вращения ω. Пушка уничтожает все объекты, на которые она сейчас направлена. Помогите судну определить порядок уничтожения пиратов за $\mathcal{O}(n^2)$, чтобы не допустить непотребства.

7. Шаблоном будем называть строку, состоящую из букв латинского алфавита и символов “?” и “*”. Будем говорить, что строка s подходит под шаблон p, если в p можно заменить каждый символ ? на букву и каждый символ * на строку из латинских букв (возможно, пустую) так, что результат будет равен s. Для строки s и шаблона p определите, подходит ли строка под шаблон за $\mathcal{O}(|s| \cdot |p|)$. Подсчитайте количество способов, которыми строка подходит под шаблон.

8. Даны два шаблона a и b, состоящие из букв латинского алфавита и символов “?” и “*”. Найдите какое-либо слово минимальной длины, подходящее под оба шаблона, за время $\mathcal{O}(|a| \cdot |b|)$.

9. Пусть дан взвешенный орграф на n вершинах, $n \leq w$ (w — разрядность машинного слова).

   (a) Найдите кратчайший гамильтонов путь за $\mathcal{O}(n^2 2^n)$.

(b) Найдите кратчайший гамильтонов цикл за $\mathcal{O}(n^2 2^n)$.

(c) Проверьте существование гамильтонова пути за $\mathcal{O}(n 2^n)$.

10. Есть k грузовиков с заданной вместимостью, задача – перевезти n вещей с заданными весами минимальным числом заездов. Один заезд – погрузить и отправить все грузовики.

    (a) $k = 1$, $\mathcal{O}(3^n)$.

    (b) $k = 2$, $\mathcal{O}(4^n)$.

    (c) $\mathcal{O}(3^n k)$.

7.2 Домашнее задание

1. *(только группа Кораблинова)* Дан выпуклый n-угольник. Каждая диагональ n-угольника, соединяющая вершины i и j, имеет вес w_{ij}. Вес триангуляции многоугольника есть сумма весов диагоналей, которые в ней проведены. Найти триангуляцию с минимальным весом за $\mathcal{O}(n^3)$.

2. Дана строка s длины n. Для каждой пары (i, j) найти длину максимального общего префикса i-го и j-го суффиксов строки s. $\mathcal{O}(n^2)$.

3. Дан набор нечестных монеток с вероятностью выпадения орла $p_1, p_2, \ldots, p_n$. Требуется посчитать вероятность выпадения ровно k орлов за $\mathcal{O}(n \cdot k)$. Операции над числами считать выполнимыми за $\mathcal{O}(1)$.

4. Пусть есть n подарков разной натуральной стоимости и три поросёнка. Нужно раздать подарки как можно честнее (так, чтобы минимизировать разность суммарной стоимости подарков самого везучего поросёнка и самого невезучего). Придумайте алгоритм решения данной задачи за $\mathcal{O}(nW^2)$, где W — суммарная стоимость подарков.

5. Шаблоном будем называть строку, состоящую из букв латинского алфавита и символов “?” и “*”. Будем говорить, что строка s подходит под шаблон p, если в p можно заменить каждый символ ? на букву и каждый символ * на строку из латинских букв (возможно, пустую) так, что результат будет равен s.

   (a) *(кроме группы Лапенка)* Для строки s и шаблона p определите, подходит ли строка под шаблон за $\mathcal{O}(|s| \cdot |p|)$.

   (b) Подсчитайте количество способов, которыми строка подходит под шаблон, по простому модулю за $\mathcal{O}(|s| \cdot |p|)$. *Подсказка: можно использовать такие же состояния динамики, как и в пункте (a)*

6. *(кроме группы Кораблинова)* Дана строка из латинских букв длины n, нужно ее запаковать в максимально короткую, используя правило (k, i) — повторить k символов начиная с i-й позиции. Заметим, что длина (k, i) – не константа. Например, $xyabababababz \to xyab(8, 2)z$, $xyaaaabaaaabaaaab \to xya(3, 2)b(10, 2)$ (но это не оптимально, оптимально $xyaaaab(10, 2)$).

   (a) $\mathcal{O}(n^3)$.

   (b) $\mathcal{O}(n^2)$, считая, что длина строки (k, i) – константа.

   (c) $\mathcal{O}(n^2)$.

Дополнительные задачи

7. У профессора есть k яиц и n этажное здание. Он хочет узнать максимальное x: если яйцо бросить с x-го этажа, оно не разобьётся. Не разбившиеся яйца можно переиспользовать. Минимизировать число бросков в худшем случае.

   (a) $\mathcal{O}(n^2 k)$.

   (b) $\mathcal{O}(nk \log n)$.

   (c) $\mathcal{O}(nk)$.

   (d) $\mathcal{O}(n \log n)$.

8 Архидемоническое программирование

8.1 Практика

1. Даны две последовательности длины n. Придумайте, как найти наидлиннейшую общую подпоследовательность этих последовательностей за $\mathcal{O}(n \log n)$, в случае, если в одной из последовательностей все элементы различны.

2. Дано рекуррентное соотношение:

$$a_n = a_{n-1} + 2b_{n-1}$$
$$b_n = 10b_{n-1} - a_{n-1}$$

   Найти a_n, b_n, c_n по модулю p за $\mathcal{O}(\log n)$.

3. Найти максимальное по весу паросочетание за $\mathcal{O}(n)$ на
   (a) дереве из n вершин,
   (b) простом цикле из n вершин,
   (c) связном неориентированном графе из n вершин и n рёбер.

   Веса на рёбрах.

4. Судно атакуют пираты. Для каждого пирата известны его азимут a_i и время t_i, через которое пират приплывет и совершит непотребство. Однако, у судна есть лазерная пушка, которой оно защищается. У пушки есть начальный азимут a и угловая скорость вращения ω. Пушка уничтожает все объекты, на которые она сейчас направлена. Помогите судну определить порядок уничтожения пиратов за $\mathcal{O}(n^2)$, чтобы не допустить непотребства.

5. Пусть дан взвешенный орграф на n вершинах, $n \leq w$ (w — разрядность машинного слова).
   (a) Найдите кратчайший гамильтонов цикл за $\mathcal{O}(n^2 2^n)$.
   (b) Найдите кратчайший гамильтонов путь за $\mathcal{O}(n^2 2^n)$.
   (c) Проверьте существование гамильтонова пути за $\mathcal{O}(n 2^n)$.

6. Посчитайте количество простых циклов в неорграфе за $\mathcal{O}(n^2 2^n)$.

7. Есть k грузовиков с заданной вместимостью, задача – перевезти n вещей с заданными весами минимальным числом заездов. Один заезд – погрузить и отправить все грузовики.
   (a) $k = 1$, $\mathcal{O}(3^n)$.
   (b) $k = 2$, $\mathcal{O}(4^n)$.
   (c) $\mathcal{O}(3^n k)$.

8. Вычислите, сколькими способами можно замостить доминошками клетчатое поле
   (a) $n \times 3$, за время $\mathcal{O}(n)$.
   (b) $n \times m$, за время $\mathcal{O}(4^n m)$.
   (c) $n \times m$, за время $\mathcal{O}(2^n nm)$.

   Ответ посчитать по модулю небольшого простого числа.

9. Клетки поля $n \times 5$ покрашены в чёрный и белый цвета. Будем называть получившийся узор красивым, если он не содержит одноцветного квадрата 2×2. Вычислите число красивых узоров по модулю небольшого простого числа за время $\mathcal{O}(\log n)$.

8.2 Домашнее задание

1. Дана строка из латинских букв длины n, нужно ее за $\mathcal{O}(n^3)$ запаковать в максимально короткую, используя правило $n(S) = \underbrace{SS\ldots S}_{n}$.

   Например `UNIXYESYESYESUNIXYESYESYES` $\to$ `2(UNIX3(YES))`.

2. Дан циклический массив, перед i-м идёт элемент $(i-1) \bmod n$, после i-го идёт $(i+1) \bmod n$. Рассмотрим операцию `zero(i) { ans += a[next(i)]*a[i]*a[prev(i)], a[i] = 0 }` Найти последовательность операций, максимизирующую ans.

   (a) $\mathcal{O}(n^3)$.

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

3. В Галлии есть n деревень, некоторые из которых соединены дорогами, длины которых – натуральные числа. Известно, что от каждой деревни до любой другой можно добраться единственным способом. Астерикс и Обеликс решили устроить праздничный забег в честь очередной победы над подразделением Цезаря. Они хотят, чтобы забег начинался и заканчивался в деревнях, маршрут не проходил ни по какой деревне дважды и имел длину ровно m. Помогите им посчитать количество различных возможных маршрутов за $\mathcal{O}(nm)$.

4. *(только группа Кораблинова)* Пусть дан взвешенный орграф на n вершинах, $n \le w$ (w — разрядность машинного слова).

   (a) Найдите кратчайший гамильтонов путь за $\mathcal{O}(n^2 2^n)$. *Подсказка: добавьте в состояние динамики еще один параметр, кроме маски.*

   (b) Найдите кратчайший гамильтонов цикл за $\mathcal{O}(n^2 2^n)$.

   (c) Проверьте существование гамильтонова пути за $\mathcal{O}(n 2^n)$. *Подсказка: элементарная операция с двумя машинными словами выполняется за* $\mathcal{O}(1)$.

5. *(кроме группы Кораблинова)* Есть k грузовиков с заданной вместимостью, задача – перевезти n вещей с заданными весами минимальным числом заездов. Один заезд – погрузить и отправить все грузовики.

   (a) $k = 1$, $\mathcal{O}(3^n)$. *Подсказка: можно делать переходы сразу по подмаске.*

   (b) $\mathcal{O}(3^n k)$. *Подсказка: научитесь узнавать, можно ли увезти подмножество вещей за 1 заезд.*

6. *(только группа Мишунина)* Найдите во взвешенном графе гамильтонов цикл минимального веса, который удовлетворяет дополнительно следующему свойству: сначала номера посещенных вершин возрастают, а затем убывают. Время $\mathcal{O}(n^2)$.

Дополнительные задачи

7. Найти максимальное по весу паросочетание на кактусе за $\mathcal{O}(n)$. Кактус — граф, в котором каждое ребро лежит не более чем на одном простом цикле.

8. Даны n предметов. У каждого есть цена v_i и вес w_i. Каждый предмет можно взять от 0 до k_i раз. Найти максимальную суммарную цену, которую можно набрать таким образом, чтобы суммарный вес не превышал W. $\mathcal{O}(nW)$.

9. Есть n вещей, у каждой есть стоимость v_i и вес w_i. Есть рюкзак, в котором можно унести набор вещей суммарного веса не более W за один подход. За $m = 2^k$ подходов унести вещи максимальной суммарной стоимости. Время $\mathcal{O}(3^n k)$.

10. Дано поле $n \times n$ с дырками. Сколько способов замостить его фигурами 1×3?

11. Вам дана доска фанеры размера $n \times m$. В нее было вбито несколько гвоздей с целыми координатами (от них остались некрасивые дырки). Сколькими способами можно разрезать доску на прямоугольники с целыми сторонами так, чтобы ни один из гвоздей не попал внутрь прямоугольника. Время: $\mathcal{O}(n^2 4^m)$.

8

9 Жадные алгоритмы

9.1 Практика

1. Постройте код Хаффмана за $\mathcal{O}(n)$, если частоты букв уже даны в отсортированном порядке.
2. В фирму поступило n заказов, которые можно выполнять в произвольном порядке. На выполнение заказа i необходимо время t_i. В каждый момент времени можно работать ровно над одним заказом. Пусть e_i – момент окончания выполнения заказа номер i. Распределите работу над заказами так, чтобы минимизировать $\sum_i e_i$. Время $\mathcal{O}(n \log n)$.
3. Есть n работ, у каждой есть t_i — время ее выполнения и d_i — дедлайн.
   (a) Определите, можно ли выполнить их все. $\mathcal{O}(n \log n)$.
   (b) Найдите максимальное подмножество, которое можно выполнить. $\mathcal{O}(n \log n)$.
4. Даны n гномов. Если i-го гнома укладывать спать a_i минут, он потом спит b_i минут. Можно ли сделать так, чтобы в какой-то момент все гномы спали? $\mathcal{O}(n \log n)$.
5. Машина тратит единицу топлива на километр, имеет бак объема k и находится в начале прямой дороги в точке 0. Для всех $i \in \mathbb{N} \cup \{0\}$ в i километрах от неё есть заправочная станция со своей положительной ценой c_i. Определите за время $\mathcal{O}(n)$, как проехать n километров за минимальную стоимость.
6. Даны n монеток, у каждой есть своя вероятность выпадения орла p_i. Нужно выбрать подмножество размера k из них, для которого вероятность выпадения ровно $\frac{k}{2}$ орлов при одновременном подбрасывании максимальна (k — четное).
   (a) $\mathcal{O}(n \log n + k^3)$.
   (b) $\mathcal{O}(n + k^2)$.
7. Маршрутка совершает рейс от первой до n-й остановки. В маршрутке m мест для пассажиров. Есть k человек, про каждого заранее известно, что он хочет доехать от остановки s_i до f_i. Проезд для пассажира стоит 1 вне зависимости от расстояния между остановками. Максимизируйте прибыль, при условии, что можно выбирать, кого сажать в маршрутку на каждой остановке. $\mathcal{O}((n + m + k) \log m)$.
8. Будем называть *независимым множеством* или *антикликой* попарно несвязное подмножество вершин графа. Пусть в графе G есть n вершин, а максимальная степень равна d. Найдите в нём независимое множество размера хотя бы
   (a) $\frac{n}{d+1}$ за время $\mathcal{O}(n)$
   (b) $\sum\limits_{v \in V(G)} \frac{1}{\deg(v)+1}$ за время $\mathcal{O}(n \log n)$

   Считайте, что граф уже дан в памяти в виде массива, где для каждой вершины хранится список её соседей.
9. Имеется n деталей и два станка. Каждая деталь должна сначала пройти обработку на первом станке, затем — на втором. При этом i-ая деталь обрабатывается на первом станке за a_i времени, а на втором — за b_i времени. Каждый станок в каждый момент времени может работать только с одной деталью. Требуется составить такой порядок подачи деталей на станки, чтобы итоговое время обработки всех деталей было бы минимальным. $\mathcal{O}(n \log n)$.

9.2 Домашнее задание

1. (a) Пусть в тексте встречается символ с частотой большей $\frac{2}{5}$. Доказать, что код Хаффмана будет содержать слово длины 1.

   (b) Пусть в тексте все символы встречаются реже $\frac{1}{3}$. Доказать, что при кодировании Хаффмана любое кодовое слово будет иметь длину не меньше 2.

2. Преподаватели сделали n заявок на занятие. Каждое занятие начинается в момент b_i и кончается в момент e_i (занимает интервал $[b_i, e_i)$). Два занятия в одной аудитории быть не могут. Распределите заявки по аудиториям так, чтобы общее число аудиторий было минимально. Решить за $\mathcal{O}(n \log n)$.

3. n атлетов хотят выстроить из своих тел башню максимальной высоты. Башня это цепочка атлетов, первый стоит на земле, второй стоит у него на плечах, третий стоит на плечах у второго и т.д. Каждый атлет характеризуется силой s_i и массой m_i. Сила это максимальная масса, которую атлет способен держать у себя на плечах. Известно, что если атлет тяжелее, то он и сильнее, но атлеты равной массы могут иметь различную силу. Определите максимальную высоту башни. $\mathcal{O}(n \log n)$.

4. Есть n человек. Человек i готов примкнуть к нашей банде, если наш авторитет хотя бы a_i, при этом он изменит наш авторитет на b_i. Наш изначальный авторитет равен A. $a_i, b_i, A \in \mathbb{Z}$

   (a) Можем ли завербовать всех людей? $\mathcal{O}(n \log n)$.

   (b) Какое максимальное число людей мы можем завербовать? $\mathcal{O}(n \log n)$.

Дополнительные задачи

5. **Пятачок, у тебя есть дома ружье?**
   Даны n непересекающихся кругов на плоскости. Мы стоим в точке $(0, 0)$ и можем стрелять по прямой. Минимальным числом выстрелов проткнуть все круги.

6. n школьников упали в яму глубины S. Каждый школьник имеет рост (от ног до плеч) h_i и длину рук l_i. Школьники могут вставать друг другу на плечи, верхний школьник может вытянуть руки. Чтобы выбраться из ямы необходимо дотянуться руками до уровня земли.

   (a) Могут ли выбраться все школьники? $\mathcal{O}(n \log n)$.

   (b) Какое максимальное число школьников может выбраться? $\mathcal{O}(n \log n)$.

7. В фирму поступают заказы, которые можно выполнять в произвольном порядке. В каждый момент времени можно работать ровно над одним заказом. Изначально заказов нет, i-й заказ поступает в момент времени r_i, работать над ним нужно t_i. Пусть e_i – момент окончания выполнения заказа номер i. Распределите работу над заказами так, чтобы минимизировать $\sum_i e_i$. Переходить от одного заказа к другому можно в любой момент времени (даже если заказ не доделан до конца, незаконченный заказ можно будет возобновить с того же места). Свойства заказа (r_i, t_i) не известны до момента его поступления. Время $\mathcal{O}(n \log n)$, при условии, что всего поступит n заказов.

8. *Раскраской вершин* графа $G = (V, E)$ называется функция $c : V \to [m]$, сопоставляющая каждой вершине G цвет от 1 до m. Раскраска называется *правильной*, если каждая пара соседних вершин имеет разные цвета.

   Для неориентированного графа $G = (V, E)$ его *хроматическим числом* $\chi(G)$ называется наименьшее возможное число цветов в правильной раскраске G.

   Для графа G обозначим размер его максимального полного подграфа через $\omega(G)$.

   Рассмотрим на вещественной прямой замкнутые отрезки $I_1, I_2, \ldots, I_n$. Сопоставим каждому отрезку I_i вершину v_i и каждой паре пересекающихся отрезков (I_i, I_j) ребро (v_i, v_j). Такой граф будем называть *интервальным графом*.

Будем называть граф G *совершенным*, если для любого его индуцированного подграфа H верно $\omega(H) = \chi(H)$.

Докажите, что каждый интервальный граф совершенен. Приведите алгоритм, красящий интервальный граф $G = (V, E)$ с $|V| = n$ в $\omega(H)$ цветов за время $O(n \log(n))$.

9. В фирму поступают заказы, которые можно выполнять в произвольном порядке. В каждый момент времени можно работать ровно над одним заказом. Изначально заказов нет, i-й заказ поступает в момент времени r_i, работать над ним нужно t_i.

   Все заказы объединены в проекты (один заказ относится к одному проекту, заказы из одного проекта могут поступать не подряд).

   Пусть e_i – момент окончания выполнения последнего (в порядке выполнения) из заказов в проекте с номером i.

   Нужно распределить работу над заказами так, чтобы минимизировать $\sum_i e_i$. Свойства заказа (r_i, t_i, его проект) не известны до момента его поступления. Переходить от одного заказа к другому можно в любой момент времени (даже если заказ не доделан до конца).

   Придумайте решение, которое не более чем в два раза хуже оптимального. Время $\mathcal{O}(n \log n)$, при условии, что всего поступит n заказов.

10 Остовные деревья

10.1 Практика

1. Пусть дан взвешенный связный неорграф $G = \langle V, E\rangle$ с выделенной вершиной s. Все веса положительны и различны. Могут ли какое-либо минимальное покрывающее дерево в G и какое-либо дерево кратчайших путей из s не иметь ни одного общего ребра? Если да, приведите пример. Если нет, докажите, что такого не может быть.

2. В стране n аэропортов. Самолет может сделать перелет из аэропорта i в аэропорт j, израсходовав w_{ij} горючего. При этом $w_{ij} = w_{ji}$, и $w_{ii} = 0$. Требуется найти минимальный размер бака, позволяющий добраться самолету из любого города в любой, возможно с дозаправками. Решить за $\mathcal{O}(n^2)$.

3. Рассмотрим следующий алгоритм поиска минимального покрывающего дерева (алгоритм Борувки):

   - Пока в графе больше одной вершины:
     - Для каждой вершины найдем самое легкое инцидентное ей ребро и добавим его в множество S (одно и то же ребро может быть выбрано дважды).
     - Добавим все ребра из множества S в ответ.
     - Стянем граф по ребрам из S.

   Докажите, что такой алгоритм найдет минимальное покрывающее дерево в случае, если веса всех ребер в графе различны, при этом время работы будет $\mathcal{O}(E \log V)$.

   Придумайте, как модифицировать алгоритм, если возможны равные веса.

4. (a) Постройте пример для алгоритма Борувки, на котором он делает $\Theta(\log n)$ фаз.
   (b) Постройте пример для алгоритма Борувки, на котором он делает $\Theta(1)$ фаз.

5. Найти во взвешенном неорграфе такой цикл, что максимальный вес ребра этого цикла минимален. $\mathcal{O}((V + E) \log E)$.

6. Дан взвешенный связный неориентированный граф $G = \langle V, E\rangle$ и некоторое минимальное остовное дерево на нём. Пусть некоторого ребра $e \in E$ изменился вес. По графу, остовному дереву, ребру e и его новому весу найдите новое минимальное остовное дерево за $\mathcal{O}(V + E)$.

7. Пусть даны взвешенный неорграф с неединственным минимальным остовным деревом и какой-то из его минимальных остовов. Найдите минимальный остов графа, отличный от данного. $\mathcal{O}(E \log V)$.

8. Пусть все ребра графа имеют различный вес. Докажите, что минимальное покрывающее дерево единственно.

9. Проверить, что минимальное по весу остовное дерево единственно. $\mathcal{O}(E \log V)$.

10. Найдите за полиномиальное время второе по весу остовное дерево в неографе.

11. Дан взвешенный орграф, постройте ориентированное к корню остовное дерево с корнем в вершине 1 минимального веса.

10.2 Домашнее задание

1. Верно ли, что для любых двух минимальных остовных деревьев в фиксированном графе наборы весов рёбер в них совпадают?

2. *(кроме группы Лапенка)* По неориентированному графу и его минимальному остовному дереву найдите второе остовное дерево (т.е. самое легкое остовное дерево из несовпадающих с данным в условии минимальным; деревья сравниваются как множества ребер) за время $\mathcal{O}(V^2 + E)$.

3. Пусть дан связный взвешенный неорграф, будем рассматривать его ребра в порядке невозрастания веса и удалять текущее ребро, если связность графа при этом не нарушается. Докажите, что этот алгоритм находит минимальный остов, или придумайте контрпример.

4. *(кроме группы Кораблинова)* Докажите, что максимальный вес ребра на пути между парой вершин в минимальном остове графа G не зависит от выбора конкретного минимального остова G.

5. Есть n городов. Можно соединить два города дорогой, потратив $A * len$ денег, где len – длина дороги, а можно построить аэропорт, потратив B денег. Нужно за минимальное число денег соединить города (чтобы от каждого до каждого можно было добраться с помощью дорог и самолетов)

6. Вам нужно передать с одного компьютера на другой n файлов. Каждый файл – битовый вектор размера m. Файл можно передать либо просто как последовательность битов, либо как $diff$ с другим файлом, уже пересланным ранее. В первом случае надо будет передать m бит, во втором - $A + B * d$ бит, где d - число различающихся битов, A и B – некоторые константы. Найти минимальное число битов, которое нужно передать, чтобы отправить все файлы с одного компьютера на другой.

Дополнительные задачи

7. **Второе остовное дерево lvl. 2**
Дан взвешенный связный неорграф G, вес его минимального остова равен w. Найдите за полиномиальное время минимальный по весу среди таких остовов G, вес которых строго превосходит w.

8. Дан взвешенный граф с положительными весами, в вершинах его стоят числа. Требуется доставить все числа в вершину 0. За то, чтобы провести число a по ребру веса w надо заплатить сумму aw. Если два числа находятся в одной вершине, то их можно слить, заменив на максимум. Постройте алгоритм доставки, дающий константное приближение к оптимальному ответу за полиномиальное время.

11 Система непересекающихся множеств

11.1 Практика

1. Ребра только добавляются, online, после каждого добавления говорить, является ли граф двудольным.

   (a) $\mathcal{O}(m \log n)$.

   (b) $\mathcal{O}(m \log^* n)$.

2. В каждой клетке прямой записано число 0 или 1. Поступает информация: четность числа единиц на отрезке $[L_i, R_i]$, найти первый запрос, после которого данные противоречивы. $\mathcal{O}(m \log^* n)$

3. Дано корневое дерево из n вершин. Все ребра ориентированы к корню. Путь называется вертикальным, если его вершина-конец является предком вершины-начала. На рёбрах дерева есть веса. Даны m вертикальных путей, за $\mathcal{O}((n+m)\log n)$ времени и $\mathcal{O}(n)$ дополнительной памяти с найти минимум на каждом из путей.

4. Пусть при реализации СНМ (классической, на деревьях) использовалось только сжатие путей. Покажите, что время обработки m запросов get можно оценить как $\mathcal{O}((m+n)\log n)$. Подсказка: рассмотрите отдельно легкие и тяжелые ребра. Ребро называется легким, если размер поддерева его нижнего конца меньше, чем половина размера поддерева его верхнего конца, тяжелым — иначе.

5. Дан набор из n корневых деревьев, каждое из одной вершины. Операции online:

   - подвесить за корень одно из деревьев к другому ребром заданного веса,
   - посчитать сумму весов ребер на пути из вершины a в ее предка b (гарантируется, что запросы поступают только про пары вершина-предок).

   (a) $\mathcal{O}(\log n)$ на операцию.

   (b) $\mathcal{O}(\log^* n)$ на операцию.

6. Бюрократы ставят печати. В компании есть n бюрократов. Сверху спустили поручение выполнить m заданий в заданном порядке. Задания могут быть трех типов.

   - Назначить x начальником y (гарантируется, что у y до этого начальства не было).
   - Поручить x проставить печать на документе i, где i – номер запроса. После этого x передает документ начальнику, тот ставит печать, передает начальнику и т.д. Самый главный начальник ставит печать и скармливает документ шредеру.
   - Узнать, ставил ли x печать на документе i.

   Поскольку все документы порезаны на мелкие кусочки, на вас последняя надежда. Нужно ответить на все запросы за $\mathcal{O}(m \log^*(n))$ offline.

7. Докажите, что в СНМ на n вершинах m последовательных запросов выполняются за $O(m \log^{**} n) + O(n)$. Подсказка: раньше рассматривались ребра двух типов: крутые ($rank_e > x^{rank_b}$) и обычные. Теперь так же рассмотрим "очень крутые" ребра — ребра, которые были крутыми в процессе сжатия хотя бы $rank_b$ раз.

11.2 Домашнее задание

Определение. *Мы будем рассматривать алгоритмы с запросами двух типов:*

- *На каждый online-запрос требуется ответить сразу после его поступления за определённое время.*
- *На каждый offline-запрос требуется ответить после поступления всех запросов, время дано на все запросы вместе.*

1. Дан неориентированный граф с весами на вершинах, в котором в начальный момент нет ребер. Нужно online обрабатывать следующие запросы:

   - соединить ребром пару вершин.
   - по вершине узнать размер связной компоненты, которой вершина принадлежит, и самую легкую вершину в этой компоненте.

   Амортизированное время на запрос — $\mathcal{O}(\log^* V)$.

Дополнительные задачи

2. Бюрократы ставят печати. В компании есть n бюрократов. Сверху спустили поручение выполнить m заданий в заданном порядке. Задания могут быть трех типов.

   - Назначить x начальником y (гарантируется, что у y до этого начальства не было).
   - Поручить x проставить печать на документе i, где i – номер запроса. После этого x передает документ начальнику, тот ставит печать, передает начальнику и т.д. Самый главный начальник ставит печать и скармливает документ шредеру.
   - Узнать, ставил ли x печать на документе i.

   Поскольку все документы порезаны на мелкие кусочки, на вас последняя надежда. Нужно offline ответить на все запросы за $\mathcal{O}(m \log^* n)$.

3. В компании работают бюрократы, один из которых является директором. У каждого бюрократа, кроме директора, есть непосредственный начальник. Приходят запросы:

   - Уволить бюрократа x. Если у него были подчиненные, то их новым начальником становится начальник x. Уволить директора нельзя.
   - Принять на работу x и назначить его непосредственным начальником y. Гарантируется, что y еще работает.
   - Отправить x сообщение. Если x уже уволен, то сообщение переадресовывается сотруднику, бывшему в момент увольнения начальником x (процесс повторяется, пока не найдтся еще работающий сотрудник). В ответ на запрос нужно вывести сотрудника, который прочитает сообщение.

   В начальный момент в компании есть один директор. Поступает n запросов, на каждый из которых нужно отвечать online за $\mathcal{O}(\log^* n)$ в среднем.

4. Дан взвешенный неориентированный граф и m запросов вида «по паре вершин и числу x узнать, существует ли между ними путь, проходящий только по ребрам не тяжелее x». Нужно ответить на все запросы offline за время $\mathcal{O}(E \log V + m \log m + m \log^* V)$.

5. Пусть при реализации СНМ (классической, на деревьях) использовалось только сжатие путей. Покажите, что время обработки m запросов *get* можно оценить как $\mathcal{O}((m + n) \log n)$. Подсказка: рассмотрите отдельно легкие и тяжелые ребра. Ребро называется легким, если размер поддерева его нижнего конца меньше, чем половина размера поддерева его верхнего конца, тяжелым — иначе.

12 DFS

12.1 Практика

1. Даны два множества вершин: A и B, за $\mathcal{O}(V+E)$ проверить, есть ли путь из какой-нибудь вершины $a \in A$ в какую-нибудь вершину $b \in B$.

2. Ориентировать неорграф так, чтобы он стал ацикличным за $\mathcal{O}(V+E)$.

3. Дано подвешенное дерево, нужно с линейным предпосчётом научиться отвечать на запрос "правда ли вершина u лежит в поддереве вершины v" online за $\mathcal{O}(1)$.

4. Проверить, является ли данный неориентированный граф двудольным $\mathcal{O}(V+E)$.

5. Дано корневое дерево и m телепортов. Для каждой вершины v дерева насчитайте самую высокую вершину, куда можно телепортироваться из поддерева v.

6. Дан связный неорграф, нужно найти в нем за $\mathcal{O}(V+E)$ все

   (a) мосты.

   (b) точки сочленения.

7. Ориентировать неорграф так, чтобы он стал сильно связным за $\mathcal{O}(V+E)$.

8. Дан DAG,

   (a) найти в нем гамильтонов путь за $\mathcal{O}(V+E)$.

   (b) проверить единственность топологической сортировки за $\mathcal{O}(V+E)$.

9. (a) Найти цикл в орграфе через данное ребро за $\mathcal{O}(E)$.

   (b) Найти цикл в орграфе через данную вершину за $\mathcal{O}(E)$

   (c) Найти цикл в неорграфе через данное ребро за $\mathcal{O}(E)$.

   (d) Найти цикл в неорграфе через данную вершину за $\mathcal{O}(E)$.

   (e) Найти в неорграфе какой-нибудь цикл за $\mathcal{O}(V)$.

10. В стране n аэропортов. Самолет может сделать перелет из аэропорта i в аэропорт j, израсходовав $w_{ij} > 0$ горючего. При этом w_{ij} может отличаться от w_{ji}, и $w_{ii} = 0$. Требуется найти минимальный размер бака, позволяющий добраться самолету из любого города в любой, возможно с дозаправками. Решить за $\mathcal{O}(n^2 \log n)$.

11. Найти в орграфе все вершины, через которые проходит любой путь $a \rightsquigarrow b$. $\mathcal{O}(V+E)$.

12. За $\mathcal{O}(V+E)$ найти в неорграфе какой-нибудь цикл

    (a) нечетной длины.

    (b) четной длины.

13. У каждой вершины не более 3-х врагов. Разбить на 2 доли так, чтобы с вершиной в долю попало не более 1 врага. $\mathcal{O}(V+E)$.

12.2 Домашнее задание

1. Пусть дано дерево $T = \langle V, E\rangle$. Для каждого ребра вычислить, сколько путей проходит через него. Придумать алгоритм, который работает за линейное время.

2. Корневое дерево T на V вершинах задается массивом из V элементов. Все вершины пронумерованы. Для каждой вершины в массиве указан его родитель. Для корня r значение в массиве равно -1. Требуется определить, как будет выглядеть новое представление дерева, если корень r сменить на корень q. Разрешается использовать $\mathcal{O}(1)$ дополнительной памяти. Менять массив можно. Время $\mathcal{O}(V)$.

3. Найти в **неорграфе** все вершины, через которые проходит любой путь $a \rightsquigarrow b$. $\mathcal{O}(V + E)$.

4. Дано дерево $T = \langle V, E\rangle$.

   (a) Найдите центроид дерева — вершину c, для которой минимальна $\sum_{v\in V} dist(v, c)$. $\mathcal{O}(|V|)$.

   (b) Найдите 2-центроид дерева — пару вершин (a, b), для которой минимальна $\sum_{v\in V} \min(dist(v, a), dist(v, b)))$. $\mathcal{O}(|V|^2)$.

   (c) Найдите 2-центроид за $\mathcal{O}(|V|)$.

Дополнительные задачи

5. Найти в **орграфе** все вершины, через которые проходит любой путь $a \rightsquigarrow b$. $\mathcal{O}(V + E)$.

6. Для заданного ориентированного графа G посчитать минимальное число ребер, которые нужно добавить в граф, чтобы он стал сильно связным. Время $\mathcal{O}(V^3)$.

13 BFS

13.1 Практика

1. Даны три бочки на 10, 7 и 4 литра. 7- и 4-литровые бочки заполнены водой. Разрешается переливать воду между бочками, либо пока одна бочка не заполнится полностью, либо пока другая не опустеет. Проверить, можно ли получить 2 литра в любой из бочек. Если возможно, найти минимальное необходимое число переливаний. Как решить аналогичную задачу, если емкости бочек a, b и c ($a, b, c \in \mathbb{N}$, $a \leq b \leq c$), а бочки с ёмкостями a и b заполнены водой? Время: $\mathcal{O}(a \cdot b)$.

2. Найти количество путей (необязательно простых) в графе за $\mathcal{O}(V^3 \log k)$
   (a) между всеми парами вершин длины ровно k.
   (b) между парой вершин длины $\leq k$.

3. Найти в орграфе все вершины, через которые проходит какой-нибудь кратчайший путь $a \rightsquigarrow b$. $\mathcal{O}(V + E)$.

4. Пусть в графе есть ребра веса 0 и 1. Придумайте, как найти кратчайшее расстояние от вершины s до остальных за $\mathcal{O}(V + E)$.

5. Пусть в графе все ребра имеют целый вес из $[1, k]$. Придумайте, как найти кратчайшее расстояние от вершины s до остальных.
   (a) $k = 2$, $\mathcal{O}(V + E)$
   (b) k — произвольное, $\mathcal{O}(V + kE)$.
   (c) k — произвольное, $\mathcal{O}(kV + E)$.
   (d) k — произвольное, $\mathcal{O}(E \log k)$.

6. Дан орграф с положительными весами на ребрах. Найти кратчайший путь, проходящий по всем k выделенным вершинам. Время $\mathcal{O}(2^k E \log (2^k V))$.

7. Пусть длина пути определяется как сумма весов всех ребер по модулю n. Найти кратчайший путь за $\mathcal{O}((V + E)n)$.

8. В стране n аэропортов. Самолет может сделать перелет из аэропорта i в аэропорт j, израсходовав $w_{ij} > 0$ горючего. При этом w_{ij} может отличаться от w_{ji}, и $w_{ii} = 0$. Требуется найти минимальный размер бака, позволяющий добраться самолету из любого города в любой, возможно с дозаправками. Решить за $\mathcal{O}(n^2)$.

9. Даны две параллельных прямых (река). В реке есть n островов (точек). Мы хотим провести по реке корабль, представляющий собой открытый круг радиуса R, так, чтобы он не задел ни одного острова. Найти максимальный R, при котором это еще возможно, за $\mathcal{O}(n^2)$.

10. (a) k-$2k$-bfs, веса вещественные.
    (b) k-$2k$-bfs, веса целые, хотим делать только целочисленные операции.

11. Есть n проводов. Есть круг из $2n$ разъемов, каждый разъем имеет тип от 1 до n (соответствующий номеру провода, который можно туда воткнуть), каждый тип встречается ровно два раза. У каждого провода есть цвет. В целях безопасности нельзя втыкать провода одинакового цвета в соседние разъемы. Найти способ соединить каждый из n проводов с одним из двух подходящих разъемов, не нарушив правила на цвета соседних.

13.2 Домашнее задание

1. В орграфе есть бесплатные и K типов платных ребер. Передвижение по любому ребру стоит 0, но, чтобы двигаться по платному ребру типа x, необходимо иметь пропуск того же типа. В любой момент времени можно иметь не более одного пропуска, однако в любой вершине есть возможность купить пропуск любого типа за A и продать пропуск любого типа за B ($0 < B < A$, цены одинаковы во всех вершинах). Найдите самый дешевый способ добраться из вершины s до вершины t за $\mathcal{O}(K(V+E))$.

2. Дан ориентированный граф с выделенными вершинами s и t. Перемещение по любому ребру занимает 1 единицу времени. В начальный момент времени мы стоим в вершине s, и во все ребра с постоянной скоростью L начинает поступать вода. У каждого ребра e есть емкость C_e — минимальное количество воды в этом ребре, при котором по нему больше нельзя двигаться (мы должны закончить движение по ребру не позднее момента достижения его емкости). Найдите максимальное L, при котором все еще существует возможность добраться от s до t за $\mathcal{O}((V+E)\log C_{\max})$.

3. Постройте матрицу $n \times n$, состоящую из клеток-стенок и пустых клеток, на которой при запуске `BFS` из какой-то клетки максимальный размер очереди будет $\omega(n)$. Ходить можно между клетками, смежными по стороне.

Дополнительные задачи

4. Модифицируем условия задачи про бочки: есть n бочек, i-я бочка имеет объем v_i ($v_i > 0$) и в начальный момент содержит c_i воды ($0 \le c_i \le v_i$). Операция переливания определена как в задаче из практики. Придумайте такие n и вещественные v_i, c_i, чтобы соответствующий граф состояний, достижимых с помощью переливаний из начального, получился бесконечным.

5. Дан набор степеней $d_1, d_2, \cdots, d_n$. Требуется построить граф $G = \langle V, E \rangle$, чтобы степень вершины v_i была равна d_i, или сказать, что такого не существует. Граф не должен содержать петли и кратные ребра. Решить за $\mathcal{O}(V+E)$.