

3 Шустрая сортировка и порядковые статистики

3.1 Практика

```
1 def qsort(a):
2     n = len(a)
3     if n <= 1:
4         return a
5     lx = []; mx = []; rx = []
6     x = a[randint from [0, n]]
7     for v in a:
8         if v < x:
9             lx.append(v)
10        elif x < v:
11            rx.append(v)
12        else:
13            mx.append(v)
14    return qsort(lx) + mx + qsort(rx)
```

1. (!) **Какую сортировку применить?**

Инверсией в массиве чисел $a[...]$ 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 $\mathcal{O}(n \log n)$.

2. (!) **Задача о флаге Нидерландов**

Робот Иван Семёныч 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. У робота есть лишь $\mathcal{O}(1)$ памяти. Помогите Ивану Семёнычу отсортировать пирожки по типу: сначала первый тип, потом второй, потом третий. Время работы — $\mathcal{O}(n)$.

3. (!) **Inplace QuickSort3**

Пользуясь предыдущей задачей, постройте inplace-версию QuickSort, то есть такую, которая не использует дополнительных массивов.

```
1 def qsort3_inplace(a, l, r): # сортируем полуинтервал a[l, r)
2     if r - l <= 1:
3         return
4     x = a[randint from [l, r]]
5
6     # Partition
7     # ...
8     # a[l:i) < x
9     # a[i:j) == x
10    # a[j:r) > x
11
12    qsort3_inplace(a, l, i)
13    qsort3_inplace(a, j, r)
```

4. (*) **k -я порядковая статистика детерминированно**

- (а) Постройте детерминированный (то есть без использования случайных битов) алгоритм поиска k -й порядковой статистики с гарантированным временем работы $\mathcal{O}(n)$. Докажите оценку на время работы.
- (б) Постройте детерминированную версию алгоритма **QuickSort** с гарантированным временем работы (то есть временем работы в худшем случае) $\mathcal{O}(n \log n)$. (Такой алгоритм получается с большой константой, поэтому непрактичен.)

5. **Опять частичная сортировка**

Дан массив длины n , выдать в порядке возрастания наименьшие k элементов за $\mathcal{O}(n + k \log k)$. (Другими словами, выдать префикс длины k отсортированной версии массива.)

Заметим, что в прошлый раз мы научились решать эту задачу за $\mathcal{O}(n + k \log n)$ с помощью кучи.

- 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 $i + 1$ ($\forall i \in [1, k - 1]$). $\mathcal{O}(n \log k)$ в среднем.
- 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. 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 -ю порядковую статистику.

3.2 Домашнее задание

1. **(1)** 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). $\mathcal{O}(n \log n)$ в среднем.
2. **(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.
3. **(1.25)** 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) **(0.75)** По позиции в отсортированном массиве.

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

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

- (b) **(0.5)** По значению.

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

4. **(*) (+1)** Даны два массива из положительных целых чисел a и b , размер обоих равен n . Выбрать массив p из k различных чисел от 1 до n так, чтобы $\frac{\sum_{i=1}^k a_{p_i}}{\sum_{i=1}^k b_{p_i}} \rightarrow \max$. Время $\mathcal{O}(n \log M)$, где $M = \max(\max(a_i), \max(b_i), n)$. Подсказка: используйте бинарный поиск по ответу.
5. **(0.5 + 1)** Дан массив из $2n$ различных чисел.
 - (a) **(0.5)** Найдите минимальное и максимальное за $3n - 2$ сравнения.
 - (b) **(*) (+1)** Докажите, что это точная нижняя оценка, то есть меньшего количества сравнений может не хватить.