Линейный поиск k-й порядковой статистики. Преобразование Фурье

1. Дано n точек плоскости, заданных своими координатами (x_i, y_i). Предложите как можно более быструю процедуру нахождения круга минимального радиуса с центром в начале координат, содержащего не менее половины точек. (Считаем, что арифметические операции и сравнения выполняются за единицу времени.)

2. Рассмотрим детерминированный алгоритм поиска порядковой статистики за линейное время из параграфа 9.3 Кормена. Какая асимптотика будет у алгоритма, если делить элементы массива на группы по три, а не по пять?

3. Предложите асимптотически эффективный алгоритм (с точки зрения количества сравнений), который получает на вход массив a_i из n различных чисел и число $k < n/2$ и выдает сумму

$$\sum_{i=1}^{k}(a_{(i)} + a_{(n+1-i)}),$$

где $a_{(i)}$ — i-ая порядковая статистика.

4. Вы хотите перемножить многочлены $A(x) = x^3+3x^2+2x+1$ и $B(x) = 2x^2+1$ с помощью БПФ. Вычислите БПФ для многочлена $A(x)$, которое можно будет использовать для дальнейшего умножения.

5. Вычислить (быстрое) обратное преобразование Фурье вектора $[6,\ -1-i,\ 4,\ -1+i]$.

6. Решите с помощью преобразования Фурье задачу о поиске всех вхождений слова (образца) в текст. Текст и образец — это последовательности $t_0, t_1, \ldots, t_{n-1}$ и $p_0, p_1, \ldots, p_{m-1}$, $m < n$ над конечным алфавитом. Образец входит в текст в позиции $i \in \{0, \ldots, n-m-1\}$, если $t_{i+j} = p_j$ при всех $j \in \{0, \ldots, m-1\}$. Для решения этой (и более сложной задачи в домашнем задании) есть $O(n \log n)$ алгоритм, основанный на БПФ. Закодируем каждый символ алфавита уникальным числом и определим последовательность r_i:

$$r_i = \sum_{j=0}^{m-1} \left(p_j - t_{i+j}\right)^2 = \sum_{j=0}^{m-1} \left(p_j^2 - 2p_j t_{i+j} + t_{i+j}^2\right).$$

1. Докажите, что образец входит в текст в позиции i тогда и только тогда, когда $r_i = 0$.

2. Постройте $O(n \log n)$ алгоритм, который находит все вхождения образца в текст.

Заметка. Эта задача подготовлена на основе статьи P. Clifford, R. Clifford Simple deterministic wildcard matching, Information Processing Letters, Vol. 101, Is. 2, 2007, Pp. 53-54,